

The Future of Programming

Human interactions with AI Coding Bots

Candidate number: 42439

Course code:MG4C3

Word count: 1542

Lecturer: Dr Will Ventures

Table of Contents

Introduction.....	2
Literature Review.....	2
AI Enhancing Job Roles.....	2
Autonomous Code Generation.....	2
Next-Generation AI Coding Assistants	3
Users in Machine Systems and Actor-Network Theory (ANT).....	3
Democratization of Coding.....	4
The Evolution of Programming	4
Historical Development in Coding and Coding Interfaces.	4
Current State of AI in Coding and The Future of AI Tool Use	4
The Changing Medium of Knowledge Exchange.....	5
Future Predictions and Conclusion	6
References.....	7

Introduction

This essay examines the current and future methods of human interaction with digital technologies, focusing on the evolution of how humans code, interact and build digital technology. The overarching argument is that AI agents will become so proficient at coding that human coders could become irrelevant, and AI agents become the most prolific method of coding, designing, and building digital technology and tools. AI agents and coding tools have already gotten good enough to replace the traditionally human tasks in programming. (Zhang, 2024). AI agents are evolving to fully assist programmers in writing code, debugging, and suggesting structural improvements (Muller et al., 2024). Traditionally, humans have used coding languages and virtual development environments to create and manipulate digital constructs. However, with the advent of AI agents, these standardized techniques are poised for a transformative change. We will now examine the literature surrounding the topic.

Literature Review

This literature review explores the significant and popular academic papers on the theme of human and technology interaction, focusing on AI coding tools. These papers provide contemporary insights into academic aspects of AI tools such as usability, the socio-technical implications and changing human roles.

AI Enhancing Job Roles

The human's role in the development process is still going to be vital, the innovation, curation and design are still areas humans outperform AI Coding Tools. Singh (2024) states "The narrative of AI fully replacing humans oversimplifies the complex interplay between technology and human labour". AI is augmenting coding roles, not replacing them. The roles of software engineers will evolve, with a greater emphasis on overseeing AI systems, ensuring data quality, and integrating AI into software development processes (Abu Zohair, 2018).

Autonomous Code Generation

The ability for full suite coding bots to create digital tools and constructs has enabled end-to-end solutions of low code entrepreneurs to create technology.

AI tools and agents are capable of autonomously generating code, taking high-level descriptions provided by developers and transforming that description into functional software. (Akhoro & Yildirim, 2025). These AI agents circumvent the traditional processes for coding software and overall improves the accessibility of technology with these tools. There is no universal trust in these tools. Programmers report they actively avoid using Large Language Models (LLMs) to practice independent learning and are wary of the quality of AI-generated code," (Akhoro & Yildirim, 2025b).

Next-Generation AI Coding Assistants

In "Envisioning the Next-Generation AI Coding Assistants: Insights & Proposals," the research focuses on the impact of AI-assisted coding tools on software development practices. This sought to measure the change in human behaviour and job roles and responsibilities and mentioned that the training of these AI tools was a huge determining factor in their ability to code. (Nghiem et al., 2024). The quality of AI-generated output heavily depends on the training data. Ensuring clean data collection, purity and preprocessing techniques is essential to maintain the accuracy of AI tools. (Siedsma, 2024). The quality of your data dictates the quality of your entire system.

Users in Machine Systems and Actor-Network Theory (ANT)

The complex interactions between human and technology in certain papers are described as essential for designing human interfaces for interacting with technology. Understanding the human role and their common behaviours allows the designing of better AI tools. Agarwal et al (2024) said in support of this point "Understanding user dynamics in machine learning systems is essential for designing effective and user-friendly AI tools".

The Actor-Network Theory (ANT) framework that is used to examine the human technology relationship by examining Actors in interactions. ANT has been used to highlight the interactions between human and non-human actors in coding environments (Aka, 2019). This is a powerful tool in measuring the impact of the changing relationship between human and AI actors in the technology creation process. We will use this later to explore the human technology relationship in the Discussion section.

Democratization of Coding

Zhang (2024) discusses the democratisation of coding through AI tool development, potentially enabling non-programmers to create digital tools and platforms through AI coding tools or agents. AI-powered development tools will continue to be developed to the point programmers will rely on them not just for code completion, now also insights into project requirements, bug detection, and design decisions. Developers will input high-level requirements and get a complete solution that meets the needs of the project" (Zhang, 2024).

The Evolution of Programming

Historical Development in Coding and Coding Interfaces.

The evolution of programming has always been transitioning and changing, growing from physical circuits, binary code to higher-level languages and modern AI tools (O'Reilly, 2025). Early coding was done in assembly, where each instruction corresponded directly to a specific machine instruction, making it efficient but difficult to understand and code. The 1970s, the introduction of the C programming language and compilers during this period allowed programmers to write in high-level languages without worrying about the underlying machine code.

The 1980s and 1990s introduced object-oriented programming languages like C++ and Java, which made it easier to manage complex software projects. Languages like Python, Ruby, and PHP emerged as powerful scripting languages and version control systems like GitHub revolutionized collaborative coding. Now AI-powered tools like GitHub Copilot and OpenAI Codex have emerged, leveraging models to assist with code completion, error detection, and generating entire code snippets based on natural language descriptions.

Current State of AI in Coding and The Future of AI Tool Use

New AI tools like GitHub Copilot and GPT-4 enhance coding efficiency and reduce errors, already reducing the amount of coding developers do in their roles. AI tools and agents can receive traditional coding languages and human language inputs and translate them into code. They empower innovators to bring their ideas to life without extensive coding knowledge.

As coding becomes more accessible, skills like creativity, problem-solving, and digital literacy become increasingly important. AI is not replacing jobs, but augmenting them, allowing jobs to adapt and focusing on replace automated tasks (O'Reilly, 2025).

The Changing Medium of Knowledge Exchange

However, with the advent of AI, code is becoming increasingly optimized for machine efficiency, making it less interpretable by humans. Many modern jobs in programming are babysitting roles managing algorithms that achieve results with unknown computations, humans only understanding the output, not how it reached it. As more of our digital technology becomes more powerful and its computation more abstract, Humans will need another layer of technology to function as an intermediary. The increasing complexity and optimization of AI-generated code may lead to a loss of human readability. (Alenezi & Akour, 2025)

This shift could result in humans relying more on AI to understand and interact with code. As AI coding bots become more advanced, humans may lose the ability to code independently, relying on AI for translation and execution. This shift represents a change in the medium or vehicle of interaction with technology, where humans would rely on AI to produce code. Not programming in digital coding environments, the coding environment of the future would become chat boxes or vocal interactions. These systems would function as intermediaries, translating human instructions into machine-optimized code.

This potential dependence raises questions about the future of human agency in coding. The reliance on AI tools brings concerns of the ethical and social implications like issues of bias, accountability, and fateful erosion of human unpredictability and authenticity.

Actor Network Theory can help us understand the power dynamics, translation processes, and ethical implications of this relationships. (Aka, 2019) The Actors would be Human programmers and AI Agent coding bots. The Network is the coding environment where humans and AI bots interact, but also the language they will speak, human language like English or Mandarin. The active translation would be the AI coding bots translating human language into code and building the construct. Power Dynamics of the AI coding bots gaining leverage by automating tasks, reduce the need for human coders, and potentially making them irrelevant or dependant. Yet these tools cannot act without us, hopefully balancing the scale.

Conclusion

We have tracked the early development of coding, examined the literature on AI coding tool and we have also examined a potential future for human technology interactions. We explored why AI coding bots could become the standard method for coding in the future. The accessibility of coding through these bots will become standardized, leading to more people interacting and building technology through these tools. You could still see programmers manually code in specific circumstances, but this would be the minority.

The transformative impact of AI coding bots will have on the future of programming will be hard to fathom. Many changes will occur over the next five years, requiring programmers to proactively build new skills and focus on curation and creation. AI coding bots will revolutionize the field of programming, making coding more accessible and efficient, but also raising important ethical and social questions about the future of human agency in coding.

With AI agents become increasingly proficient, the human coder's role and responsibilities may diminish, leading to a new era where AI intermediaries dominate the landscape of digital development. The emergence of AI coding bots will redefine human-technology interaction, making it more intuitive and efficient. As these AI agents continue to advance, they will bridge the gap between human language and technology, ushering in a new era of digital development.

References

- Abu Zohair, L. M. (2018). (PDF) The Future of Software Engineering by 2050s: Will AI Replace Software Engineers? *ResearchGate*.
https://www.researchgate.net/publication/330185339_The_Future_of_Software_Engineering_by_2050s_Will_AI_Replace_Software_Engineers
- Agarwal, A., Eliassi-Rad, T., Heidari, H., & Lazaric, A. (2024). *Humans, Algorithmic Decision-Making and Society: Modeling Interactions and Impact*.
<https://icml.cc/virtual/2024/workshop/29971>
- Aka, K. G. (2019). Actor-network theory to understand, track and succeed in a sustainable innovation development process. *Journal of Cleaner Production*, 225, 524–540. <https://doi.org/10.1016/j.jclepro.2019.03.351>
- Akhoroz, M., & Yildirim, C. (2025). *Conversational AI as a Coding Assistant: Understanding Programmers' Interactions with and Expectations from Large Language Models for Coding* (No. arXiv:2503.16508). arXiv.
<https://doi.org/10.48550/arXiv.2503.16508>
- Alenezi, M., & Akour, M. (2025). AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Applied Sciences*, 15(3), Article 3.
<https://doi.org/10.3390/app15031344>
- Muller, M., Weisz, J. D., Houde, S., & Ross, S. I. (2024). *Drinking Chai with Your (AI) Programming Partner: Value Tensions in the Tokenization of Future Human-AI Collaborative Work*. <https://dl.acm.org/doi/fullHtml/10.1145/3663384.3663390>
- Nghiem, K., Nguyen, A. M., & Bui, N. D. Q. (2024). *Envisioning the Next-Generation AI Coding Assistants: Insights & Proposals* (No. arXiv:2403.14592). arXiv.
<https://doi.org/10.48550/arXiv.2403.14592>
- O'Reilly, T. (2025). *The End of Programming as We Know It*. O'Reilly Media.
<https://www.oreilly.com/radar/the-end-of-programming-as-we-know-it/>
- Siedsma, A. (2024). *Will AI Replace Programmers? Navigating the Future of Coding*.
<https://extendedstudies.ucsd.edu/news-events/extended-studies-blog/will-ai-replace-programmers-navigating-the-future-of-coding>

- Singh, Dr. V. K. (2024). Human Intelligence in the Age of AI: Why Machines Won't Take Over Jobs. *IJSRET*. <https://ijsret.com/2024/09/25/human-intelligence-in-the-age-of-ai-why-machines-wont-take-over-jobs/>
- Varma, R. (2025, March 8). *How AI Tools Are Reshaping the Coding Workforce*. CXO Magazine. <https://magazine.cxo.net/latest-trends/technology/how-ai-tools-are-reshaping-the-coding-workforce>
- Zhang, D. (2024). AI-Powered Coding Tools: A Study of Advancements, Challenges, and Future Directions. *Science and Technology of Engineering, Chemistry and Environmental Protection*, 1(10), Article 10. <https://doi.org/10.61173/n7m7qa37>