Green coding – bridging the gap from theory to practice

Verena Majuntke, Lena Hoffmann, Felix Kronlage-Dammers, Grischa Beier, and Nikolas Becker

Abstract: This article summarises key points from the interdisciplinary panel discussions held at the Weizenbaum Conference 2024 titled "Green Coding – Bridging the Gap from Theory to Practice", where experts shared diverse perspectives on the importance of Green Coding (GC) in sustainable software development. As digital technologies and services expand, their ecological costs rise, with data centres consuming 1.0–1.5% of global electricity in 2022 and information and communication technologies (ICT) contributing about 1.5–4% of global greenhouse gas (GHG) emissions. The panel highlighted the role of software in influencing the energy and resource consumption of hardware. GC focuses on enhancing energy and material efficiency across the software lifecycle, addressing the growing demands of ICT driven by cloud computing, artificial intelligence and video streaming. Despite some companies recognising the importance of GC, implementation remains low and is often not part of established practices in software development. The discussion emphasised the need for open-source initiatives, standardised metrics, industry integration, stakeholder awareness and robust governance to align digital transformation with sustainability goals. This article highlights the main arguments and insights from the panel discussions.

Keywords: sustainable software, energy-efficient software, green coding, software-induced resource consumption, open source

1. Green coding – bridging the gap from theory to practice

The transition towards sustainable development requires a critical examination of the environmental emissions caused by digital technologies and services, as well as counteracting strategies to mitigate these impacts. **Green Coding (GC)** has emerged as a vital approach to enhancing the energy efficiency of software. Based on an interdisciplinary panel discussion at the Weizenbaum Conference 2024, this article elaborates on the current state of GC, the challenges in its measurement and the role of open-source methodologies in promoting sustainable software development.

2. Why is GC important for the transformation towards sustainability development?

Digital technologies and services are a vital part of increasingly numerous areas of social life; however, digital innovations also come with increasing ecological costs. In 2022, the operation of data centres already accounted for approximately 1.0–1.5% of global electricity demand (IEA, 2023). However, estimates regarding global energy consumption by information and communication technologies (ICT) vary greatly, particularly due to differing definitions of ICT (Gelenbe, 2023). Studies such as Hilty et al. (2009) argue that there is no clear definition of the scope of ICT; for example, they consider three building blocks of ICT: servers, end-user devices and the networking infrastructure typically used to communicate either between two or more end-user devices or among end-user devices and servers. As a result of differing underlying assumptions, estimates regarding ICT's share of global greenhouse gas (GHG) emissions also vary greatly, with systematic reviews estimating emissions at 1.8%–2.8% (Freitag et al., 2021) and 1.5%–4% (Bieser et al., 2023). One major part of ICT-related emissions is determined in the production phase of ICT hardware. For end-user devices, these emissions often exceed 50% of the device's overall life cycle GHG emissions, while for data centres, it is estimated that most of the life cycle-related GHG impacts are caused during the use phase (Bieser et al., 2023).

In contrast to hardware, concepts aimed towards making software usage more resource efficient have historically seen lower prioritisation. However, software is the core of ICT, and ICT is continuously growing and evolving to make software more efficiently available for users, i.e. industry and private consumers. While technologies become more efficient, the gained efficiency tends to be utilised to transmit greater volumes of data rather than making the transmission of existing data more resource-efficient, a phenomenon known as the **Jevons Paradox**. It is necessary to focus more extensively on the energy and material efficiency of data infrastructure and data centres while also looking to **make software operations as resource-efficient as possible**. Combining these approaches is an important factor and a crucial prerequisite if we want to be successful in achieving a more general transition towards sustainable development.

GC is one approach aimed at increasing the energy efficiency of software usage. GC is defined as "**the act of designing, developing, maintaining, and (re-)using software systems in a way that requires as little energy and natural resources as possible.**" Green Coding methods or practices thus mean any action or use of technology intended and suitable to further this (Junger et al., 2024). Moreover, GC concerns itself with the entire lifecycle since software may have a long history, with its resource consumption not limited to the version currently in use.

Against the backdrop of the rapidly growing number of data centres and volumes of data being hosted – reasons for which include but are not limited to the increasing usage of cloud computing, artificial intelligence and video streaming – there is an urgent demand to optimise the resource consumption of the respective applications and the data operation processes of data centres. GC may be a promising concept for both of these issues, as it helps guide practitioners to become more conscious about the material and energy consumption of the systems they design and use. All hardware and software are tied to inevitable resource demands, and as long as they are associated with an environmental footprint, their usage needs to be carefully considered.

3. What is the current situation concerning GC?

There exists a substantial body of scientific research addressing the concept of sustainable software. Research in the context of GC assesses a broad range of aspects, including the energy requirements of different programming languages (Pereira et al., 2017), the holistic energy consumption throughout the software development process and the criteria for ensuring sustainable software development practices (Gröger et al., 2018). Additionally, there has been an increased focus on the energy consumption of machine learning models at various stages of their lifecycle (Kaack et al., 2022; Luccioni et al., 2022; Strubell et al., 2019). However, despite the focus on sustainability in a variety of other sectors, the IT industry has largely been overlooked and has not yet prioritised sustainable software development. A study evaluating the extent to which GC practices are used in the IT industry revealed that, while half of the surveyed software companies consider GC a strategic goal, less than 17% of these companies have consciously integrated GC strategies into their operations (Henze et al., 2023).

To advance towards more sustainable software, it is essential to educate developers. Higher education institutions must ensure that graduates are aware of sustainability challenges and know how to address them. Moreover, sustainability should be treated as a cross-sectional issue. Even with well-trained developers, resource-efficient software development will not be achieved unless it becomes a (sub-)target within companies. If companies and customers prioritise new features over addressing technical debts that impact quality – as can be seen in many software projects – practices aimed at improving resource efficiency will not be implemented.

A significant challenge also concerns the current lack of transparency regarding software resource consumption. In order to optimise software towards resource efficiency, resource consumption needs to be measured. Since software typically comprises and utilises a variety of services, it is crucial for all components to disclose their resource consumption publicly. Politicians could push the issue further by implementing regulations that require transparency in resource consumption.

Currently, nearly every initiative to reduce software resource consumption faces a similar challenge: it is **methodologically demanding** to precisely measure (in the sense of directly attribute) emissions for a specific application or service. Thus, an essential first research step is to establish an empirically sound basis for measurement with widely trusted and reliable reference standards and public benchmarks. Such a foundation is crucial for the subsequent development of broadly accepted labels for software and services. Some projects, such as ECO:DIGIT, address this challenge by developing an automated assessment environment that provides transparent metrics and data on the resource consumption and CO_2 emissions of software applications. However, even though increasing resource efficiency in every resource-consuming context is absolutely desirable, efficiency gains alone will not be sufficient to achieve the necessary transformation towards circular and emission-free methods of production and consumption.

The societal perspective also needs to be cautiously considered. GC must not be used to shift the focus of public debate away from the (oftentimes critical) materiality of digital transformation. Additionally, there are various issues related to the openness and inclusivity of digital infrastructure. Another important topic in the current era of AI concerns the size of data centres; the larger the data centre, the more efficiently it can process large volumes of data, but the quest for ever-greater efficiency may create the risk of concentrating power in very few hands. Careful reflection is needed to determine the extent to which we want our access to the internet to depend on a few international actors. It is apparent that adequate governance is needed to ensure that digital transformation remains inclusive and reliable while balancing the conflicting paradigms of increasing efficiency and openness.

4. What role do open-source approaches play regarding GC?

Considering the four pillars of open source – ***Freedom to use*, *Freedom to study*, *Freedom to modify* and *Freedom to redistribute*** – the advantages of its utilisation become evident. The availability of source code not only facilitates the easy adoption of well-established patterns in related projects but also creates a basis for thorough and widespread analysis. An example of such analysis is the work done by Eugenio Capra and colleagues, who examined the use of application development environments in relation to energy efficiency (Capra et al., 2012). This analysis allows for the refinement of development paradigms and a shift towards more sustainable ways of development. Furthermore, it provides the needed transparency regarding the methods applied.

It is unsurprising that a variety of methodologies and available datasets originated from projects within the open source community, such as Boavizta. Various efforts have arisen within the broader community over the past several years, with opensustain.tech and the OSS for climate podcast providing fitting examples. Still, the power of the decentralised nature of open source is also its flip side: the vast amount of software and approaches lead to the need for standardisation and the development of best practices.

5. Advancing research and practical implementation of GC – what key steps and strategies are needed?

Following the principle "you can't manage what you can't measure", it is crucial to encourage companies to measure the environmental impact of their software. However, as stated above, the process of acquiring accurate measurements is challenging for many companies. In response to increased public awareness of climate change, international digital corporations have started to include digital emissions in their annual sustainability reports and equip their software services with tools that help customers track and assess their sustainability. These approaches, however, are based on varying assumptions and often lack **transparency** if the underlying methodologies remain undisclosed and the results cannot be verified or replicated by external parties. To address this challenge, there is an urgent need for **trusted and reliable reference standards and public benchmarking** to allow for comparability and transparency (which in turn can also help to prevent greenwashing). These standards should enable companies to measure their software's environmental impact based on transparent and scientifically sound methods. In addition, we need to advance environmental accounting and measurement methodologies, not only for the resource consumption of software but also for communication (network) and computing (data centre) infrastructure. Since the application of standards is generally voluntary and standards only become binding when included in contracts or mandated by law, policy measures are also necessary to effectively promote GC practices. Furthermore, transparency regarding applied methods can be achieved by releasing building blocks as open source, which would allow an even wider range of adoption as other methods can be built upon the same foundations and be improved together.

From a societal perspective, we need to raise awareness in all relevant stakeholder groups (developers, industries, political actors), as well as develop **governance concepts** to cope with the further centralisation (resp. concentration) of power, as environmental targets for the operation of digital technologies and ICT infrastructure can more easily be met at scale.

From an industry perspective, it is essential to examine how GC can be broadly integrated into software development. Successful case studies could demonstrate to enterprises that investing in sustainable software development is not only an ethical choice but also a cost-saving measure. While such metrics are necessary to inform economic decisions, no comparative data is currently available.

From an education perspective, sustainability must be integrated into curricula and education. Awareness that sustainability is a cross-sectional topic that needs to be addressed is necessary not only for those who actually build the software but especially for those at C-level.

In summary, to reduce the environmental impact of software, it is essential for companies to **measure** accurately, despite the challenges doing so poses. Transparent, **scientifically sound reference standards** and public **benchmarks** are urgently needed to ensure comparability and prevent greenwashing. **Policy** measures and the adoption of **open-source** methodologies can drive the broader implementation of GC practices. By integrating sustainability into **education,** industry practices and governance frameworks, we can create an **inclusive and resource-efficient digital future**.

Acknowledgements

Grischa Beier would like to thank his colleague Malte Reißig for the critical suggestions and reflections on the topic, which have enriched his work.

References

Bieser, J. C. T., Hintemann, R., Hilty, L. M., & Beucker, S. (2023). A review of assessments of the greenhouse gas footprint and abatement potential of information and communication technology. *Environmental Impact Assessment Review*, *99*, 107033. https://doi.org/10.1016/j.eiar.2022.107033

Capra, E., Francalanci, C., & Slaughter, S. A. (2012). Is software "green"? Application development environments and energy efficiency in open source applications. *Information and Software Technology*, *54*(1), 60–71. https://doi.org/10.1016/j.infsof.2011.07.005

Freitag, C., Berners-Lee, M., Widdicks, K., Knowles, B., Blair, G. S., & Friday, A. (2021). The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. *Patterns*, *2*(9), 100340. https://doi.org/10.1016/j.patter.2021.100340

Gelenbe, E. (2023). Electricity Consumption by ICT: Facts, trends, and measurements. *Ubiquity*, *2023*(August), 1–15. https://doi.org/10.1145/3613207

Gröger, J., Köhler, A., Naumann, S., Filler, A., Guldner, A., Kern, E., Lorenz, H., & Maksimov, Y. (2018). *Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik*. Umweltbundesamt.

Henze, C., Junger, D., Schauermann, E., & Weber, S. (2023). *Potentials of Green Coding: Perspectoves on implementation*. https://gi.de/en/aktuelles/projekte/en-green-coding

Hilty, L. M., Coroama, V., de Eicker, M. O., Ruddy, T. F., & Müller, E. (2009). *The Role of ICT in Energy Consumption and Energy Efficiency*. Report to the European Commission, DG INFSO, Project ICT ENSURE: European ICT Sustainability Research.

IEA. (2023). Tracking Clean Energy Progress 2023. IEA. https://www.iea.org/reports/tracking-clean-energy-progress-2023

Junger, D., Westing, M., Freitag, C. P., Guldner, A., Mittelbach, K., Obergöker, K., Weber, S.,Naumann, S., & Wohlgemuth, V. (2024). *Potentials of Green Coding—Findings and Recommendations for Industry, Education and Science—Extended Paper* (arXiv:2402.18227). arXiv. http://arxiv.org/abs/2402.18227Kaack, L. H., Donti, P. L., Strubell, E., Kamiya, G., Creutzig, F., & Rolnick, D. (2022). Aligning artificial intelligence with climate change mitigation. *Nature Climate Change*, *12*(6), 518–527. https://doi.org/10.1038/s41558-022-01377-7

Luccioni, A. S., Viguier, S., & Ligozat, A.-L. (2022). *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model* (arXiv:2211.02001). arXiv. http://arxiv.org/abs/2211.02001

Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J. P., & Saraiva, J. (2017). Energy efficiency across programming languages: How do energy, time, and memory relate? *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*, 256–267. https://doi.org/10.1145/3136014.3136031

Strubell, E., Ganesh, A., & McCallum, A. (2019). *Energy and Policy Considerations for Deep Learning in NLP* (arXiv:1906.02243). arXiv. http://arxiv.org/abs/1906.02243