COMPUTING edge

- **Data Visualization**
- **Smart Cities**
- **Mobile Computing**
- **Games**

JULY 2024

www.computer.org

IEEE COMPUTER SOCIETY

IEEE

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org

Printed with inks containing soy and/or vegetable oils

Circulation: *ComputingEdge* (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

Postmaster: Send address changes to *ComputingEdge*-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

JULY 2024 · VOLUME 10 · NUMBER 7

COMPUTING edge

Magazine Roundup

The IEEE Computer Society's lineup of 12 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to visualization and microchip design. Here are highlights from recent issues.

Computer

Performance Comparison of Software Reliability Estimation Algorithms

Specific optimization algorithms have been developed for the purpose of automated software reliability assessment tools. In this article, featured in the April 2024 issue of *Computer*, the authors propose the Monte Carlo expectation-maximization algorithm as another optimization algorithm, and carry out the performance comparison of the software reliability estimation algorithms through comprehensive numerical experiments.

Computing in Science & Engineering

Tensor Extrema Estimation Via Sampling: A New Approach for Determining Minimum/Maximum Elements

The tensor train (TT) format, widely used in computational mathematics and machine learning, offers a computationally efficient method for handling multidimensional arrays, vectors, matrices, and discretized functions in various applications. This September/October 2023 *Computing in Science & Engineering* article proposes a new algorithm for estimating minimum/maximum elements of TT-tensors, which leads to accurate approximations. The method consists of sequential tensor multiplications of the TT-cores with an intelligent selection of candidates for the optimum.

IEEE Annals of the History of Computing

Text Standards for the "Rest of World": The Making of the Unicode Standard and the OpenType Format

This January–March 2024 *IEEE Annals of the History of Computing* article traces the development of the set of technologies that enabled multilingual digital written communication, focusing on the Unicode Standard, the bottom layer of what becomes "text stack," and the OpenType Format (OTF), an important technical layer that augments the Unicode Standard. While this article considers the possibilities for multilingual text writ large, it focuses most on the family of Indic scripts.

IEEE Computer Graphics and Applications

Testing the Capability of AI Art Tools for Urban Design

The authors of this March/April 2024 *IEEE Computer Graphics and Applications* article evaluated the performance of three artificial intelligence (AI) image synthesis models, Dall-E 2, Stable Diffusion, and Midjourney, in generating urban design imagery based on scene descriptions. The results showed significant differences between the three AI models, as well as differing scores across urban scenes, suggesting that some projects and design elements may be more challenging for AI art generators to represent visually.

IEEE Intelligent Systems

Integrated Circuit Mask–Generative Adversarial Network for Circuit Annotation With Targeted Data Augmentation

In recent years, deep-learning-based segmentation techniques have been applied to circuit

2469-7087/24 © 2024 IEEE

annotation for the hardware assurance of integrated circuits (ICs). However, imperfections in circuit images often cause incorrectly segmented pixels, which result in critical circuit connection errors that are detrimental to subsequent circuit analysis. To mitigate such circuit connection errors, this article, featured in the January/February 2024 issue of *IEEE Intelligent Systems*, proposes a targeted data augmentation framework for deep-learning-based circuit annotation, termed IC Mask–Generative Adversarial Network (ICMG), which generates circuit images containing the aforementioned imperfections through generative-adversarial-network-based image translation from synthetic circuit masks.

Digital Transformation in Remote Learning and Work—An Externality of the COVID-19 Pandemic

In this January/February 2024 *IEEE Internet Computing* article, the authors review how educational institutions have digitally transformed teaching and learning processes into remote learning and how government agencies and businesses have re-engineered their business processes to facilitate remote work worldwide. Based on this review, the authors further discuss how our society will embrace these new working and learning paradigms in the post-COVID-19 era.

Compression Analysis for BlueField-2/-3 Data Processing Units: Lossy and Lossless Perspectives

This article, featured in the March/April 2024 issue of *IEEE Micro*, characterizes lossy (e.g., SZ3) and lossless (e.g., DEFLATE, lz4, and zlib) compression algorithms using seven real-world datasets on Nvidia BlueField-2/-3 DPUs. The authors explore the potential opportunities for offloading these compression workloads from the host. Their findings demonstrate that the C-engine within the DPU can achieve up to 26.8x speedup compared to its SoC core.

Retinex-Guided Channel Grouping-Based Patch Swap for Arbitrary Style Transfer

In this article, featured in the January–March 2024 issue of *IEEE MultiMedia*, the authors propose a retinex theory guided, channel grouping-based patch swap technique. A channel grouping strategy groups the style feature maps into surface and texture channels, which prevents the winner-takes-all problem. Retinex theory-based decomposition controls a more stable channel code rate generation. In addition, the authors provide a complementary fusion and multiscale generation strategy to prevent unexpected black areas and over-stylized results, respectively.

Detecting Mobile Malware Associated With Global Pandemics

A recent analysis conducted during the COVID-19 pandemic showed that several variants of COVID-19 related malware were installed by the public from non-trusted sources. The authors of this *IEEE Pervasive Computing* article, in the October–December 2023 issue, propose the use of app permissions and an extra feature (the total number of permissions) to develop a static detector using machine learning (ML) models to enable the fast-detection of pandemics-related Android malware at installation time.

IEEE SECURITY & PRIVACY

Trust in Data Security Protocols and Knowledge of Privacy and Security Technology

This article, featured in the March–April 2024 issue of *IEEE Security & Privacy*, examines the correspondence between trust in corporate data security and knowledge of the internal mechanisms companies use to guarantee privacy and security. The results show higher trust is associated with lower knowledge.

IEEE Software

Software Size Measurement: Bridging Research and Practice

The authors of this article in the May/June 2024 issue of *IEEE Software* investigate the limited adoption of functional size measurement methods in the software development industry. Using insights from firms experienced in size measurement, the article aims to uncover industry expectations and facilitate the translation of theoretical methodologies into practical applications.

IT Professional

The "X-Factor" of 6G Networks: Optical Transport Empowering 6G Innovations

The sixth generation (6G) of mobile networks will radically change the way that we interact, as the artificial intelligence (AI)-enabled fixed and wireless communications and networks infrastructure will offer improved performance and an unprecedented variety of applications. In this January/February 2024 *IT Professional* article, the authors address possible 6G innovations, such as the implementation of flexible functional splits and end-to-end network slicing, from the viewpoint of evolved x-haul networks that are based on novel and energy-efficient optical communications systems and network technologies.

Visualizing the Future of Technology

Today's technology toolbox is overflowing with brilliant tools—data visualization, AI, automation, and digital twins, to name a few. With these tools at our disposal, the question is: How best to use them? This issue of *ComputingEdge* visualizes the future of our technology use by exploring how we can apply key technologies to make our society more effective and efficient. The issue also delves into the games industry, how to get a career in gaming, and the creation of a metaverse university.

Engineers can employ data visualization to advance and support digital technology. The *Computing in Science & Engineering* article, "Data Visualization for Digital Twins," provides examples and techniques for how to use data visualization to visualize digital twins. The author of "Six Opportunities for Scientists and Engineers to Learn Programming Using AI Tools Such as ChatGPT," from *Computing in Science & Engineering*, describes a wide variety of applications for AI tools.

Smart technology offers powerful new techniques to organize and automate systems. The authors of "An Automatic and Intelligent Internet of Things for Future Agriculture," from *IT Professional*, present a mechanism using AI and data analysis to enable automated and intelligent farm cultivation management and operation.

To ensure we can keep up with advancements in mobile computing and data use, IT infrastructure must evolve. "Databiting: Lightweight, Transient, and Insight Rich Exploration of Personal Data," from *IEEE Computer Graphics and Applications*, introduces databiting as a possible system to help people gain insight from their data anywhere and anytime through small visualizations. *Computer*'s article, "Tomorrow's Applications Require IT Operations That Are Autonomous, Ubiquitous, and Smarter—In a Word, Invisible," also argues that IT infrastructure must be reinvented to meet increasing demands for computing and storage.

Aspiring artists and engineers can look to the games industry as an exciting and expanding field in which they can design future gaming and metaverse applications. In "The Metaverse University," from *Computer*, the author proposes an educational program for a university devoted to training engineers and artists to design metaverse technologies. In "How Do I Get a Position in the Games Industry? The FAQ," also from *Computer*, the author outlines steps people can take to work in the games industry, as well as qualifications and positions.

EDITORS: Rita Borgo, rita.borgo@kcl.ac.uk
Kelly Gaither, kelly@tacc.utexas.edu

This article originally appeared in computing in SCIENCE & ENGINEERING vol. 25, no. 2, 2023

DEPARTMENT: VISUALIZATION CORNER

Data Visualization for Digital Twins

João L. D. Comba, Nicolau O. Santos, Jonathan C. Rivera, Regis K. Romeu, and Mara Abel,
Federal University of Rio Grande do Sul, Porto Alegre, RS, 91501-970, Brazil

Visualization techniques are useful in the analysis and insight generation for applications in computing in science and engineering. In this article, we describe the importance of visualization to a digital twin (DT), a virtual representation of a physical object, process or system that can be applied for different tasks, such as data-driven simulation, analysis or monitoring. We illustrate tasks in DTs and give examples of how visualization techniques can be applied for DTs in different application areas.

In recent years, several areas in computational science have referred to the notion of a digital twin (DT).[1] Examples include aeronautics,[2] oil and gas,[3] agriculture,[4] engineering,[5] manufacturing,[6] healthcare,[7] and smart cities,[8] among others. The definition of a DT varies slightly depending on the application, and here we define a DT as *a virtual representation that aims to mimic a physical object, process, or system*. The intent is to replace the physical entity with a data-driven model that can be used for simulation, monitoring, prediction, and analysis.

VISUALIZATION TASKS FOR DTs

Data visualization plays a key role in DT systems as it provides ways to inspect, interact, and understand the virtual representation of a physical system. Next, we list DT tasks that can benefit from visualization techniques.

Monitoring and Analysis

An important task of a DT is to display and monitor the data the physical asset generates so that they can be analyzed to identify important events and trends. For example, visualization can display real-time data from sensors and other sources, allowing users to monitor the performance and condition of physical assets and systems. These data can be displayed in various forms, such as graphs, charts, and maps. Examples of monitoring tasks include the following:

- *Healthcare*: remote monitoring of patient health data.
- *Oil and gas*: monitoring the performance of equipment (drilling rigs, pipelines, refineries, and storage facilities) to identify production problems.
- *Aeronautics*: monitoring aircraft components in real time to improve safety and reduce downtime.
- *Smart cities*: monitoring and optimizing energy usage in buildings and infrastructure, improving energy efficiency and reducing costs.

Simulation and Optimization

Simulations are crucial to identify improvement opportunities and optimize the performance of physical assets and systems. Visualization can display simulation results, allowing users to explore scenarios in real time, test alternatives, and design more efficient solutions to problems. Examples of simulation and optimization tasks include the following:

- *Healthcare*: simulate and plan surgical procedures, and simulate the performance of medical implants.
- *Oil and gas*: simulate the flow of oil and gas through pipelines; simulate the structural stress in drilling columns, pipes, and pumps.
- *Aeronautics*: simulate and test the performance of aircraft designs, including aerodynamics, structural integrity, and fuel efficiency.
- *Smart cities*: simulate emergency situations, allowing response teams to practice and refine response plans.

1521-9615 © 2023 IEEE
Digital Object Identifier 10.1109/MCSE.2023.3295968
Date of current version 23 August 2023.

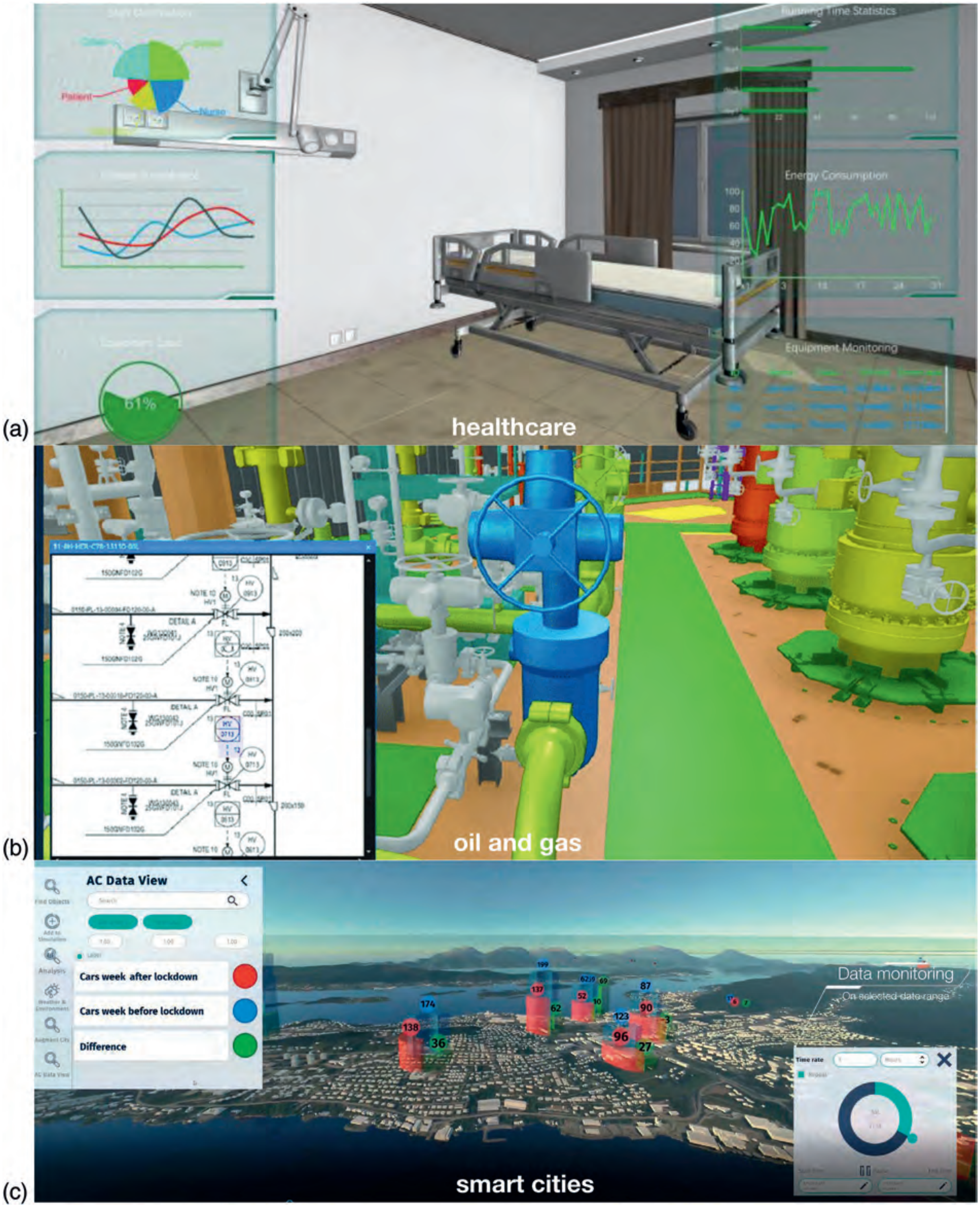

FIGURE 1. Situated visualization examples of DTs. (a) Healthcare,[7] (b) oil and gas,[9] and (c) smart cities.[10]

Predictive Maintenance and Decision Making

DTs can generate predictive maintenance alerts, highlighting potential issues before they become critical. This can be useful for maintenance teams to take proactive actions and improve the reliability and efficiency of their equipment. Visualization can help support predictive maintenance and decision-making tasks that can arise in DTs, such as the following:

- *Healthcare*: prevent illnesses like cancer or diabetes from advancing and develop better treatments.
- *Oil and gas*: identify production equipment bottlenecks, anomaly events, and areas of improvement.
- *Aeronautics*: predict both the useful life of aircraft components and their issues and schedule repairs.
- *Smart cities*: improve city planning by enabling decision makers to test different virtual scenarios and identify potential issues before implementing changes in the real world.

VISUALIZATION TECHNIQUES FOR DTs

Visualization techniques can help improve the efficiency of decision-making tasks and enhance the overall value of DT technology. Next, we enumerate examples of visualization techniques applied to DTs.

Situated Visualization and Augmented Reality

The physical structure of a DT can be described using a virtual 3-D representation that can be rendered and interactively explored. Exploring virtual scenarios allows interactions that might not be possible with physical systems. Moreover, the advance of low-cost virtual and augmented reality allows overlaying 3-D renderings with computer-generated imagery to enhance the experience and provide additional information about the DT virtual asset. Recently coined terms *situated visualization* and *situated analytics*[11] bring a new set of techniques that use mixed reality to display data relevant to a physical location directly in their location. There is great potential for using such techniques in the context of DTs.[12]

Figure 1 illustrates examples of situated analytics applications to DTs. Figure 1(a) is an application of healthcare that displays a virtual hospital room along plots that may illustrate properties of equipment's and patient's data.[7] In a diagnosis procedure, it is possible to overlay the historical patient data and real-time body parameters with the physical representation of the hospital room. Figure 1(b)[9] displays a DT for a

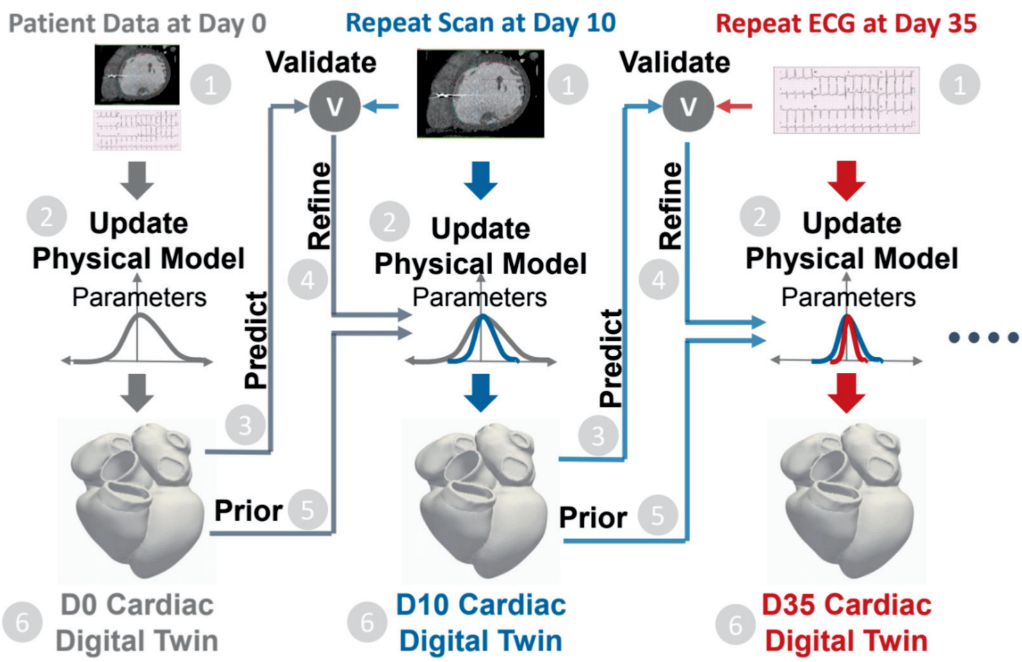

FIGURE 2. Cardiac DT. A DT for a dynamically updated physics-based heart model.[13] (Source: Steven Niederer; original figure used with permission.) D0: day zero; D10: day 10; D35: day 35.

(a)
(b)
(c)

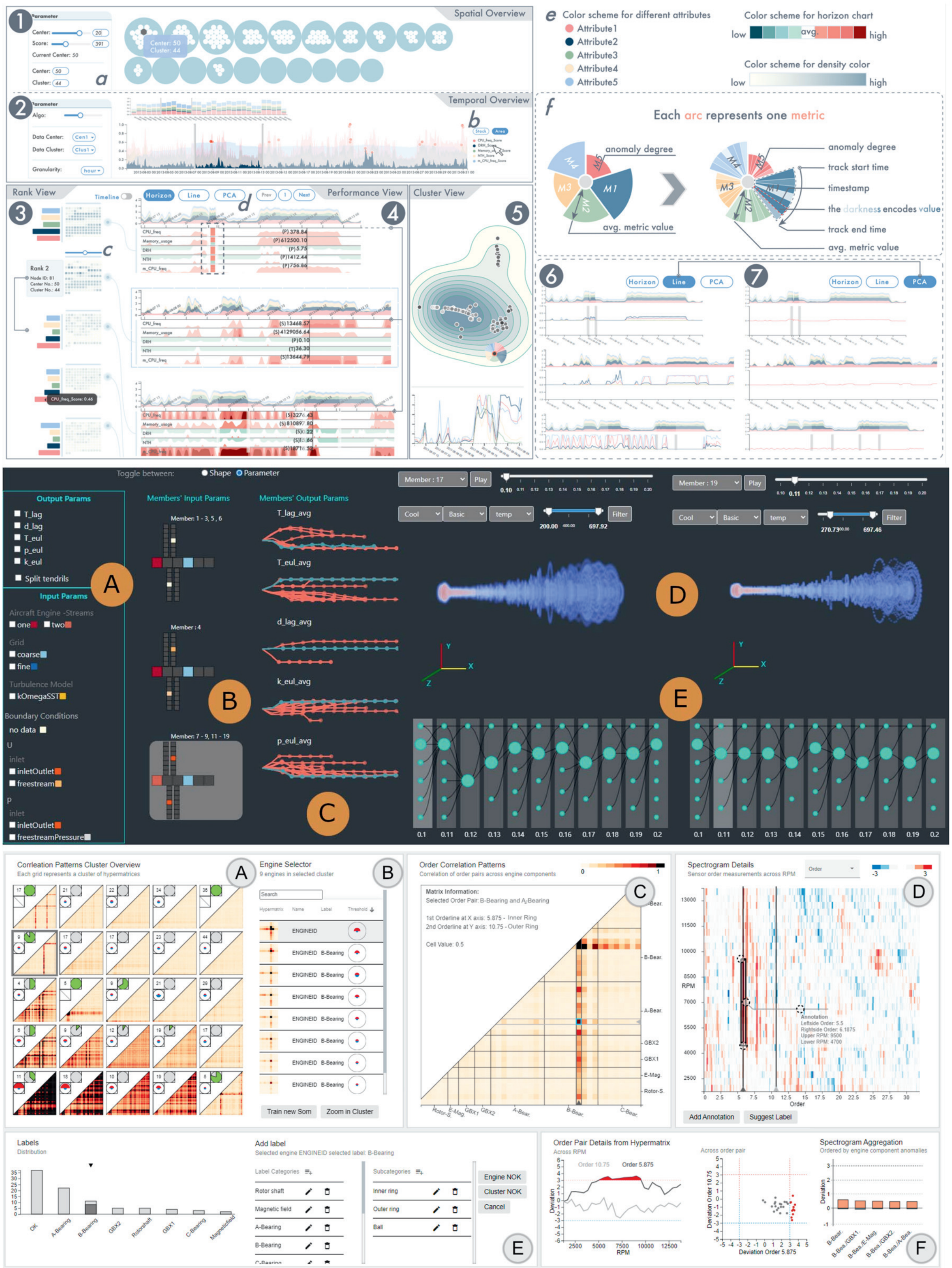

FIGURE 3. Promising visualization systems for DTs. (a) CloudDet: anomaly detection in cloud computing systems,[16] (b) a system for analyzing high-performance computing aircraft engine simulations.[17] (c) Irvine: a manufacturing system that analyzes correlation patterns of automotive electrical engines.[18]

working offshore oil platform developed by Kongsberg Digital. The figure demonstrates an overlay of 2-D process diagrams with the 3-D visualization of a production facility enriched with real-time data. Figure 1(c) depicts a case study using DTs that studied the air pollution of a city in Norway during the COVID-19 pandemic.[10] It displays a snapshot of a traffic movement animation of the smart city DT, showing a week-by-week comparison of hourly traffic before and after the lockdown.

Computational Modeling and In Situ Visualization

The increased computational performance in high-performance visualization systems has allowed computational modeling systems to start using visualization techniques inside computational pipelines, an approach referred to as *in situ visualization*.[14] Figure 2 illustrates a DT that simulates cardiac functions of the heart.[13] In this example, a patient attends a clinic and receives an electrocardiogram (ECG) on day zero (D0). The patient's ECG is given as input to a physics-based model that updates heart movements using a 3-D representation of the heart. The DT is continually refined with additional real-time ECG scans on D10 and D35 to update the physical model and the respective cardiac DT representation.

There is great potential for using in situ visualization for DTs. The technology that supports the required computational modeling is becoming more affordable, and examples of high-performance visualization systems applied to DTs have already appeared in the literature.[15]

Time-Series Monitoring and Analysis

In a DT, time-series data are typically collected from sensors and used to create a virtual model that mirrors the behavior of a physical system. Time-series analysis techniques can be applied to identify patterns, trends, and anomalies in the system's behavior. Figure 3 illustrates three recently developed systems that show great promise for use in DT systems. The focus of these works is the analysis of multivariate time-series data.

Figure 3(a) shows CloudDet,[16] a visual analytics system developed for anomaly detection in cloud computing systems. The core of CloudDet is a new anomaly detection algorithm and a rich user interface that supports anomaly analysis for multivariate time series of computational traces. The interface has a spatial view to give context about the hierarchy of the computational system and a temporal view that displays anomaly results over time. An anomaly ranking using a calendar view summarizes important events to inspect. A performance view details the data related to an anomaly selected, supporting correlation analysis of temporal patterns with different semantics and scales. Finally, a cluster view uses a multidimensional projection algorithm to ease the identification of subsets of computer nodes with similar performance.

In Figure 3(b), we illustrate a system for analyzing high performance computing aircraft engine simulations.[17] The system supports ways to summarize simulation runs that identify condensation trails (contrails) generated from emitted particles by aircraft engines. The system relies on input simulation data that pass through a computational back end that performs post-processing computations, such as reconstruction of the 3-D jet plume and computation of similar ensemble runs. The visualization interface supports filtering input data and the selection of ensembles run from a set of visual representations that encode ensemble parameters. The 3-D plume view allows for comparison of jet plumes and animated particles from similar runs. Finally, a panel displays the evolution of contrails using a node-link diagram that enables identification of merging and splitting of contrails groups.

Figure 3(c) shows the Irvine system,[18] which analyzes correlation patterns of automotive electrical engines. The input to the system is acoustic data collected through sound propagation inside engines. In a preprocessing step, the system computes the signature of an engine (called a *hypermatrix*) based on correlations of spectrograms. A self-organizing map computes clusters of hypermatrices, which are summarized in the interface along the engines in the cluster. Once a cluster and an engine are selected, details of the acoustic signature and raw acoustic measurements are displayed using line charts and scatterplots. Users can assign labels and provide annotations that can be used to document observations, such as sources of errors, which are stored in a knowledge database. The documentation of knowledge, as described in this work, can be very useful in developing DT systems.

CONCLUSION

In many ways, visualization techniques play a key role in problems related to computing in science and engineering. This article summarized examples of how visualization can be useful to DT applications.

ACKNOWLEDGMENTS

This is the farewell article for João L.D. Comba as co-editor of the "Visualization Corner" column. I, João, have been enormously satisfied playing this role since 2016. I want to thank my former Co-Editor Daniel Weiskopf and current Co-Editor Kelly Gaither for their partnership throughout the years. I also would like to thank the entire *Computing in Science & Engineering* (*CiSE*) team, particularly the current Editor-in-Chief Lorena Barba, who is an inspiration to work with, and who brought to *CiSE* a stimulating collaborative environment. As visualization continues to thrive, it will be my pleasure to follow "Visualization Corner" in the future as Rita Borgo replaces me and joins Kelly as co-editor.

REFERENCES

1. M. Singh et al., "Applications of digital twin across industries: A review," *Appl. Sci.*, vol. 12, no. 11, 2022, Art. no. 5727, doi: 10.3390/app12115727.
2. L. Li, S. Aslam, A. Wileman, and S. Perinpanayagam, "Digital twin in aerospace industry: A gentle introduction," *IEEE Access*, vol. 10, pp. 9543–9562, 2022, doi: 10.1109/ACCESS.2021.3136458.
3. T. R. Wanasinghe et al., "Digital twin for the oil and gas industry: Overview, research trends, opportunities, and challenges," *IEEE Access*, vol. 8, pp. 104,175–104,197, Jun. 2020, doi: 10.1109/ACCESS.2020.2998723.
4. W. Purcell and T. Neubauer, "Digital twins in agriculture: A state-of-the-art review," *Smart Agricultural Technol.*, vol. 3, Feb. 2023, Art. no. 100094, doi: 10.1016/j.atech.2022.100094.
5. H. Feng, C. Gomes, C. Thule, K. Lausdahl, A. Iosifidis, and P. G. Larsen, "Introduction to digital twin engineering," in *Proc. IEEE Annu. Model. Simul. Conf. (ANNSIM)*, 2021, pp. 1–12, doi: 10.23919/ANNSIM52504.2021.9552135.
6. Y. Wang, Z. Zhu, L. Wang, G. Sun, and R. Liang, "Visualization and visual analysis of multimedia data in manufacturing: A survey," *Vis. Inform.*, vol. 6, no. 4, pp. 12–21, Dec. 2022, doi: 10.1016/j.visinf.2022.09.001.
7. Z. Lv, J. Guo, and H. Lv, "Deep learning-empowered clinical big data analytics in healthcare digital twins," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, early access, 2023, doi: 10.1109/TCBB.2023.3252668.
8. H. Masoumi, S. Shirowzhan, P. Eskandarpour, and C. J. Pettit, "City digital twins: Their maturity level and differentiation from 3D city models," *Big Earth Data*, vol. 7, no. 1, pp. 1–36, 2023, doi: 10.1080/20964471.2022.2160156.
9. A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21,980–22,012, Jan. 2020, doi: 10.1109/ACCESS.2020.2970143.
10. P. Major, G. Li, H. P. Hildre, and H. Zhang, "The use of a data-driven digital twin of a smart city: A case study of Ålesund, Norway," *IEEE Instrum. Meas. Mag.*, vol. 24, no. 7, pp. 39–49, Oct. 2021, doi: 10.1109/MIM.2021.9549127.
11. S. Shin, A. Batch, P. W. S. Butcher, P. D. Ritsos, and N. Elmqvist, "The reality of the situation: A survey of situated analytics," *IEEE Trans. Vis. Comput. Graphics*, early access, 2023, doi: 10.1109/TVCG.2023.3285546.
12. A. Künz, S. Rosmann, E. Loria, and J. Pirker, "The potential of augmented reality for digital twins: A literature review," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, 2022, pp. 389–398, doi: 10.1109/VR51125.2022.00058.
13. S. Niederer, M. Sacks, M. Girolami, and K. Willcox, "Scaling digital twins from the artisanal to the industrial," *Nature Comput. Sci.*, vol. 1, no. 5, pp. 313–320, May 2021, doi: 10.1038/s43588-021-00072-5.
14. H. Childs, J. Bennett, C. Garth, and B. Hentschel, "In situ visualization for computational science," *IEEE Comput. Graph. Appl.*, vol. 39, no. 6, pp. 76–85, Nov./Dec. 2019, doi: 10.1109/MCG.2019.2936674.
15. N. S. Holliman, M. Antony, J. Charlton, S. Dowsland, P. James, and M. Turner, "Petascale cloud supercomputing for terapixel visualization of a digital twin," *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 583–594, Jan./Mar. 2022, doi: 10.1109/TCC.2019.2958087.
16. K. Xu et al., "CloudDet: Interactive visual analysis of anomalous performances in cloud computing systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 1107–1117, Jan. 2020, doi: 10.1109/TVCG.2019.2934613.
17. N. Nipu, C. Floricel, N. Naghashzadeh, R. Paoli, and G. Marai, "Visual analysis and detection of contrails in aircraft engine simulations," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 1, pp. 798–808, Jan. 2023, doi: 10.1109/TVCG.2022.3209356.
18. J. Eirich et al., "IRVINE: A design study on analyzing correlation patterns of electrical engines," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 11–21, Jan. 2022, doi: 10.1109/TVCG.2021.3114797.

JOÃO L.D. COMBA is a full professor at the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, RS, 91501-970, Brazil. Contact him at comba@inf.ufrgs.br.

NICOLAU O. SANTOS is a Ph.D. candidate at the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, RS, 91501-970, Brazil. Contact him at nicolau.santos@inf.ufrgs.br.

JONATHAN C. RIVERA is an M.Sc. student the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, RS, 91501-970, Brazil. Contact him at jcrivera@inf.ufrgs.br.

REGIS K. ROMEU is a visiting researcher at the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, RS, 91501-970, Brazil. Contact him at regisromeu@gmail.com.

MARA ABEL is a full professor at the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, RS, 91501-970, Brazil. Contact her at marabel@inf.ufrgs.br.

EDITOR: Ryan Cooper, ryan.c.cooper@uconn.edu

DEPARTMENT: EDUCATION

This article originally appeared in Computing in Science & Engineering vol. 25, no. 3, 2023

Six Opportunities for Scientists and Engineers to Learn Programming Using AI Tools Such as ChatGPT

Philip J. Guo, University of California San Diego, La Jolla, CA, 92093, USA

This article demonstrates how scientists and engineers can use modern artificial intelligence (AI) tools such as ChatGPT and GitHub Copilot to learn computer programming skills that are relevant to their jobs. It begins by summarizing common ways that AI tools can already help people learn programming in general and then presents six new opportunities catered to the needs of scientists and engineers: 1) create customized programming tutorials for one's own domain of work, 2) learn complex data visualization libraries, 3) learn to refactor exploratory code into more maintainable software, 4) learn about inherited legacy code, 5) learn new programming languages on demand within the context of one's workflow, and 6) question the assumptions that one's scientific code is making. Taken together, these opportunities point toward a future where AI can help scientists and engineers learn programming on demand within the context of their existing real-world workflows.

Over the past year (2022–2023), technology companies have released a staggering variety of artificial intelligence (AI) tools that can generate text, code, images, music, and even video clips on demand. Lately, it feels like new AI tools are coming out every week, and it is impossible to keep up with the latest buzzwords and marketing slogans. In this article, I zoom in on one popular kind of AI tool that is especially relevant for scientists and engineers: large language model (LLM)-based tools such as ChatGPT and GitHub Copilot that take text (or code) as input from the user and generate text (or code) in response. Specifically, I present six opportunities for scientists and engineers to use these LLM-based AI tools to *learn* computer programming skills that are relevant to their jobs.

As these AI tools have grown more popular, educators have started writing about ways to use them to teach and learn programming. However, most of the writing on this topic so far has been about using AI (especially within computer science departments) to teach students who aim to become full-time programmers.[1,2,3] Here, I provide a complementary perspective by addressing the unique needs of scientists and engineers who do *not* intend to become professional software developers. For instance, a climate scientist may need to pick up a bit of Python or R to analyze data for their research, or a mechanical engineer may learn command-line scripting on embedded Linux systems because they are rigging together hardware components for a measurement device. This article presents six ways that AI tools can help them learn what they need directly within the context of their existing real-world workflows.[4]

1521-9615 © 2023 IEEE
Digital Object Identifier 10.1109/MCSE.2023.3308476
Date of current version 25 October 2023.

BACKGROUND: FREQUENTLY MENTIONED THOUGHTS ABOUT USING AI TOOLS TO LEARN PROGRAMMING

Before focusing on scientists and engineers in particular, I first set the stage by summarizing what others have already mentioned about using AI to learn programming in general. I distilled these sentiments from several overview papers,[1,2,3] which each cite more detailed research studies in their respective bibliographies.

Here are some widely acknowledged capabilities of AI tools for learning programming at the time of this writing, about a year after ChatGPT was released:

- AI tools can solve many kinds of programming exercises that are now used as homework and exam questions in introductory computer science courses. As a result, some instructors are worried about students using them to cheat and are reconsidering what kinds of homework assignments and exams to give in the future.
- On a more constructive note, as AI can generate a variety of different solutions for a programming exercise, those can serve as *worked examples* (also known as *worked-out examples*[6]) that students can use to learn different approaches to solving a problem. Seeing different *variations* of solutions can help students learn better.[7]
- AI can automatically generate a wide variety of programming exercises to meet a given pedagogical goal (e.g., teaching how to join multiple data tables in Python using the Pandas library). This capability can help instructors to prepare assignments more efficiently and give students extra practice opportunities on demand.
- AI can explain what a piece of code does step by step in a novice-friendly way. This capability can benefit those students who may feel embarrassed to ask someone for help on a seemingly "simple" question; they can now freely ask the AI tool without fear of being judged.
- AI can help debug students' code, which can enable them to make progress on their homework without feeling self-conscious about asking someone for help.
- AI can automatically write tests for students' code, which can help them spot more bugs.
- AI can perform *code reviews* to give students feedback on their coding style and aesthetics.

And here are some frequently mentioned limitations of these tools:

- First and foremost, AI tools can generate code that is incorrect, buggy, insecure, or that violates other known best practices. Moreover, students may have trouble spotting the bugs in AI-generated code as it often appears to be well-written.
- Relatedly, because AI-generated code looks convincing at first glance, students may grow *overreliant*[8] and reflexively copy and paste it into their projects without questioning whether it is correct or not. As a result, instructors have emphasized that it is critical for students to learn to write test cases for AI-generated code.[2]
- Even if AI-generated code is correct and of high quality, the fact remains that the student did not write that code themselves. This makes it harder for them to understand what the code does and to make future updates to it.
- AI tools are optimized for "doing" rather than teaching. This means that when a student asks the AI tool a question, it will directly give them an answer. Although this can be convenient, it may hinder learning. In contrast, a good human tutor would teach the underlying concepts and gradually guide the student to solve the problem on their own (with well-timed hints along the way) instead of giving them the answer right away.
- It is hard for novices to develop a mental model of what these AI tools are and are not capable of because how they work is mysterious even to experts. Thus, some students may get frustrated that AI cannot help them with seemingly straightforward requests, when in reality, those requests appear vague or unclear to the AI tool. In practice, one must learn to become good at *prompt engineering*[9] to be able to write prompts (i.e., textual requests to the AI tool) that can consistently elicit high-quality responses.
- The code or explanations that AI tools produce by default may be too complex for novices to understand because AI mimics the style of code found on the Internet and is not specifically tuned to be beginner friendly.
- AI tools may generate text that reinforces existing social biases, contains toxic content, or uses copyrighted materials without the original creators' permission.

As you read about the opportunities below for AI to help scientists and engineers learn programming, please also keep the above limitations in mind.

OPPORTUNITY 1: USING AI TO CREATE CUSTOMIZED PROGRAMMING TUTORIALS FOR YOUR OWN DOMAIN

The opportunity I am most excited about is using AI tools to create programming tutorials that are customized for a scientist or engineer's own domain of work. Why is this significant? Because scientists and engineers often learn programming to analyze their own data, but existing tutorials all use *generic* datasets such as the popular "iris" or "mtcars" data that come preinstalled with the R programming language. The "iris" dataset describes 150 samples of iris flowers with features such as their sepal and petal dimensions, and "mtcars" catalogs information about 32 car models from the early 1970s, with features such as their number of carburetors, horsepower, and rear axle ratio. Unless you happen to be a flower or car enthusiast, chances are that you do not care at all about this data, so learning programming using these datasets may not be that motivating for you.

Instead of reading tutorials that use generic datasets, you can now use an AI tool like ChatGPT to generate customized tutorials with example data from your own domain of work. No matter if you are a geologist or astrophysicist or structural engineer, AI tools possess enough knowledge about the basics of your field to be able to generate synthetic data and code examples that you can relate to more than "iris" or "mtcars." For instance, if you are learning how to do multiple linear regression in Matlab, it can be far more relatable to see examples using data from your own domain (e.g., geology) rather than, say, predicting car horsepower from rear axle ratios. You can even paste in real datasets from your own research (e.g., a neuroscience study) and have the AI tool generate a programming tutorial based on your data. Learning programming using authentic data in a domain you personally care about can make that knowledge stick better than reading generic tutorials.[10] For more details, I walk through an example of this idea in the "Intermission 1: ChatGPT as a Personalized Tutor" section of Guo 2023.[4]

OPPORTUNITY 2: USING AI TO LEARN COMPLEX DATA VISUALIZATION LIBRARIES

A staple of scientific programming is writing code to produce data visualizations ranging from simple bar charts all the way to interactive multiscale dynamic diagrams. Scientists face an inevitable tradeoff here, they can either 1) use a point-and-click interface like Excel or Google Sheets or 2) write custom Python/R/JavaScript/Matlab code using libraries such as Matplotlib, Seaborn, ggplot2, Bokeh, Plotly, Altair, or D3.js. The former is easier to learn but offers less expressive power, while the latter is more expressive but harder to learn.

AI tools can lower the barriers to learning the latter category of data visualization libraries. Similar to Opportunity 1, you can use AI to generate tutorials for how to use these libraries *within the context of the data with which you are currently working*. For instance, let's say you are a marine biologist wanting to make a scatterplot to correlate observations of different types of fish, and you want the data points to vary in color, shape, and size according to certain fishy properties. Even though you may not know how to write the exact code to do so, you have a clear vision of what the output should look like. By expressing this request to an AI tool, it can both generate the data visualization code and add inline comments to teach you how that code works step by step.

Using AI to write data visualization code for you can be effective as it is something that can be hard for humans to do but easier for humans to *verify*. Let's face it, not even seasoned programmers remember how to write Matplotlib, ggplot2, or other visualization code from scratch. These complex libraries contain hundreds of different functions, each with a heap of different parameters that interact in idiosyncratic ways. It's a waste of our human brainpower to memorize all these mundane details, but AI is great at "remembering" these details for us. As we have an intuitive sense of what the output visualization should look like, we can verify whether the AI-generated code looks more or less correct and make adjustments if needed. This ease of human verification gets around a core limitation of AI tools, which is that they might generate incorrect code or output. However, note that looking at the visualization alone may not be enough to fully verify correctness, so it is still important to inspect the AI-generated code to make sure that its logic makes sense.

OPPORTUNITY 3: LEARNING TO REFACTOR EXPLORATORY CODE INTO MORE MAINTAINABLE SOFTWARE

Scientific programming is often exploratory in nature and done in a mix of creatively named files (e.g., MY_ANALYSIS_SCRIPT_v2_param53_final_FINAL.py) and computational (e.g., Jupyter or R) notebooks. The main goal, especially during early stages of research, is to iterate quickly to explore and refine hypotheses, not to produce clean, maintainable code. But if these initial explorations are successful, then inevitably, this draft code ends up living far longer than originally intended. So, one big challenge for scientists and engineers is learning to *refactor* this prototype code into something more maintainable longer term.

Refactoring is a software engineering technique where a programmer revises code to be more clear and maintainable while still preserving the same functionality. Here, AI tools can help by inspecting your code and suggesting refactoring opportunities such as creating more descriptive variable names, encapsulating common snippets into their own functions, making indentation and spacing more consistent, and adding inline comments to describe what each part of your code does. By seeing how AI refactors your code, you can learn habits that you can apply in the future. In this way, AI plays the role of a senior colleague who demonstrates best practices within the context of your own codebase. These in situ, context-specific lessons can stick better than if you had read a general guide to code refactoring.

Similar to Opportunity 2, this can be a great use case for AI as it is relatively easy for a human to verify whether the output is correct. Because you already have code that works, the AI-refactored code should behave the exact same way when you run it. You can look at the old and new versions side by side and run both to give you confidence that the AI worked as intended. Using AI to refactor can be less risky than using it to write brand-new code.

OPPORTUNITY 4: LEARNING ABOUT INHERITED LEGACY CODE

Scientists often inherit code from former lab members who have graduated or moved on to new jobs. As mentioned previously, ideally, everyone would take the time to refactor their exploratory code into something more maintainable. But in reality, lots of old code is hard to understand because it may have been duct taped together in a hurry to get experiments working for a paper submission deadline. And even if the original authors had intended to clean it up and document it well, they would always be under pressure to move on to the next project, aim for the next publication or grant deadline, and so on. Plus, we as scientists are not professional software developers, so we may lack the expertise to follow industry best practices for code quality, even if we have the best of intentions. The end result is that inherited code (formally called *legacy code*[11]) can be very hard to understand and work with, which slows down scientific progress.

Here, AI tools can help by automatically inspecting a pile of old legacy code and generating step-by-step explanations, clarifying code comments, and supplemental documentation to summarize how that code works. These explanations may not be 100% accurate, but they can serve as a starting point for human investigation. Think of the AI here as an intrepid explorer buddy who can help you out when spelunking into a deep cave of unexplored legacy code. By working alongside the AI when exploring an unknown codebase, you can learn both how to work with it specifically and also more general skills for how to effectively deal with legacy code in the future.

OPPORTUNITY 5: LEARNING NEW PROGRAMMING LANGUAGES ON DEMAND WITHIN THE CONTEXT OF YOUR WORKFLOW

Scientists and engineers may have to learn a new programming language on short notice if an important library they need is available only in that language. As they need to get their job done efficiently, they cannot easily put their main work on hold to take a formal course or work through a textbook. Instead, it would be much more convenient to be able to learn on demand within the context of their own existing workflow. AI tools can facilitate this type of *just-in-time* learning in the following two ways:

1) A scientist can write code for their task in the language they are most comfortable with (e.g., Python) and then use an AI tool to *automatically translate* it into the language they want to learn (e.g., Julia). Although this translation is by no means perfect, it is likely "good enough" to show the correspondences between the two languages (e.g., which Python features map to which Julia features). This way, someone can learn a bit of Julia directly within the context of a piece of Python code with which they are familiar.
2) Going the other way, a scientist can find a piece of example code in an unfamiliar language (e.g., Julia) and then use an AI tool to translate it back into a language they already know well (e.g., Python). This can come in handy if, say, that piece of example code implements an important algorithm that they need for their research, but they do not understand how it works because they are not familiar with the language in which it is written.

OPPORTUNITY 6: QUESTIONING THE ASSUMPTIONS YOUR SCIENTIFIC CODE IS MAKING

One of the most challenging aspects of writing scientific code is making sure that the assumptions that underlie your code are well justified. Even if you implement the most elegant, efficient, and bug-free algorithm to run on your data, if that is not the appropriate algorithm to use, then your code is still useless (or may even be harmful if it gives misleading or biased results). However, it is impossible for existing code analysis tools to tell whether your code may be making incorrect assumptions because these tools do not know anything about the underlying scientific or engineering questions your code is trying to address. Modern AI tools have the potential to overcome this limitation via a clever rhetorical trick: by asking *you* whether the assumptions you are making are sound and having you come up with answers on your own.

AI tools still cannot do your science for you, but what they can do is serve as a skeptical inquisitor to question the assumptions your scientific code is making. For instance, if you are a geneticist writing scripts using the Biopython library to process a specific type of gene sequencing data, an AI tool may know enough about this domain to ask skeptical questions about why you decided to, say, run a specific parametric statistical test, and whether that test is justified given the properties of the dataset you are using. Or the tool can question you about why you decided to use linear instead of logistic regression when your outcome variable seems to be binary. The AI tool does not necessarily know the correct answer to those questions, but it likely knows "enough" about genetics and statistical tests to pose these questions to you. This use case is like having the AI tool serve as a sort of Socratic tutor[12] to get you to introspect more deeply on your thought process.

PARTING THOUGHTS AND A CALL TO ACTION

In summary, AI tools like ChatGPT offer unique potential for scientists and engineers to learn programming while also aiding in various aspects of their work. These tools can generate personalized programming tutorials, facilitate learning of complex data visualization libraries, suggest refactoring of exploratory code, and assist in understanding inherited legacy code. Moreover, they may help in learning new programming languages in the context of work and encourage the questioning of assumptions in scientific code. However, it's essential to maintain a balanced perspective and acknowledge the limitations of these AI tools. They are only as good as their training data, can sometimes produce incorrect output, and, importantly, lack true comprehension of context and domain-specific knowledge. They should be seen as aides that can help streamline certain tasks, but are not replacements for the essential expertise, judgment, and creative problem-solving abilities of human scientists and engineers. Therefore, I encourage the scientific community to explore these tools with a critical eye, understanding their strengths and limitations. Use them where they can genuinely add value and continue relying on human intellect and insight where it matters most. In doing so, we can use technology to enhance our work without over-relying on it, striking a beneficial balance in this era of digital advancement.

I promise that the previous paragraph is the only one in this entire article that was written by an AI tool (ChatGPT with GPT-4). I asked ChatGPT to summarize what I wrote, and the original paragraph it generated was too overenthusiastic about the benefits of AI (very self-serving of it!). So then I instructed ChatGPT to "make it more balanced and less pro-AI," and it generated the more nuanced response that you just read. Although it is a well-written summary, its style doesn't really match my own. So at least for now, I will still be doing my own writing without AI assistance, even though I do use AI to help me in programming.[4] It's hard to predict how these tools will evolve in the coming years, but hopefully the ideas I presented here can serve as starting points for you to learn more about this fast-changing topic.

ACKNOWLEDGMENT

Thanks to Lorena Barba for encouraging me to write this article, Shannon Ellis for helping me brainstorm ideas for it, and Ashley Juavinett for feedback. This material is based on work supported by the National Science Foundation (NSF) under Grant NSF IIS-1845900.

REFERENCES

1. B. A. Becker et al., "Programming is hard - Or at least it used to be: Educational opportunities and challenges of AI code generation," in *Proc. 54th ACM Tech. Symp. Comput. Sci. Educ. V.1*, Mar. 2023, pp. 500–506, doi: 10.1145/3545945.3569759.
2. S. Lau and P. J. Guo, "From 'Ban It Till We Understand It' to 'Resistance Is Futile': How University programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and GitHub Copilot," in *Proc. ACM Conf. Int. Comput. Educ. Res. (ICER)*, 2023, pp. 1–16, doi: 10.1145/3568813.3600138.
3. P. Denny et al., "Computing education in the era of generative AI," *Commun. ACM*, 2023.
4. P. Guo, "Real-real-world programming with ChatGPT: Taking AI far beyond small self-contained coding tasks," *O'Reilly*, Jul. 25, 2023. [Online]. Available:

https://www.oreilly.com/radar/real-real-world-programming-with-chatgpt/
5. "Introducing ChatGPT." OpenAI. Accessed: Aug. 1, 2023. [Online]. Available: https://openai.com/blog/chatgpt
6. A. Renkl, "The worked-out examples principle in multimedia learning," in *The Cambridge Handbook of Multimedia Learning*, Cambridge, U.K.: Cambridge Univ. Press, 2005, pp. 229–246.
7. M. Thune and A. Eckerdal, "Variation theory applied to students' conceptions of computer programming," *Eur. J. Eng. Educ.*, vol. 34, no. 4, pp. 339–347, Aug. 2009, doi: 10.1080/03043790902989374.
8. S. Passi and M. Vorvoreanu, "Overreliance on AI: Literature review," Microsoft Research, Tech. Rep., 2022. https://www.microsoft.com/en-us/research/uploads/prod/2022/06/Aether-Overreliance-on-AI-Review-Final-6.21.22.pdf
9. P. Denny et al., "Conversing with Copilot: Exploring prompt engineering for solving CS1 problems using natural language," in *Proc. 54th ACM Tech. Symp. Comput. Sci. Educ. V.1*, Mar. 2023, pp. 1136–1142, doi: 10.1145/3545945.3569823.
10. M. K. Kjelvik and E. H. Schultheis, "Getting messy with authentic data: Exploring the potential of using data from scientific research to support student data literacy," *CBELife Sci. Educ.*, vol. 18, no. 2, Jun. 2019, Art. no. es2, doi: 10.1187/cbe.18-02-0023.
11. M. Feathers, *Working Effectively With Legacy Code.* Upper Saddle River, NJ, USA: Prentice-Hall, 2004.
12. E. Al-Hossami et al., "Socratic questioning of novice debuggers: A benchmark dataset and preliminary evaluations," in *Proc. 18th Workshop Innov. Use NLP Building Educ. Appl. (BEA)*, 2023, pp. 709–726, doi: 10.18653/v1/2023.bea-1.57.

PHILIP J. GUO is an associate professor of Cognitive Science with the University of California San Diego, La Jolla, CA, 92093, USA. Contact him at pg@ucsd.edu.

EDITOR: Amir Dabirian, amir@fullerton.edu

DEPARTMENT: IT TRENDS

This article originally appeared in IT Professional vol. 24, no. 6, 2022

An Automatic and Intelligent Internet of Things for Future Agriculture

Yi-Wei Ma, Jiann-Liang Chen, and Ching-Chiu Shih, *Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei City, 106, Taiwan*

Great advances in artificial intelligence and the Internet of Things have been made in recent years. This work develops an intelligent agricultural cultivation system to realize automated and intelligent farm cultivation management and operation. Experimental results demonstrate that the proposed system reduces water consumption by 50% and 30%, respectively, below those of fixed and threshold irrigation methods and predicts more accurately the amount of irrigation water required. The proposed dynamic detection transmission mechanism performs 97.8% fewer communications than are performed in timing transmission mode, effectively reducing overall energy consumption for transmission. Experimental results show that the proposed system reduces cultivation time by 40.94% below that required by the traditional cultivation method, and supports more effective management. Based on all of the experimental results, the proposed method supports convenient cultivation for growers.

MOTIVATION

The global population continues to increase as does the demand for food. Like the aging of people who are engaged in agriculture, this increase affects the human resources required in the sector. With the rapid development of automation, information and communication, Internet of Things (IoTs), Big Data and AI, agricultural production technology has been continuously developing from 1.0 to 4.0.[1]

Agriculture 1.0 is traditional agriculture, which uses substantial manpower for cultivation, which is easily affected by climate, causing the quality and variety of crops to being limited by the environment. In Agriculture 2.0, industrial technology is used, and many pieces of machinery and equipment reduce the dependence of the sector on manpower and increase the efficiency of the use of arable land, thereby increasing overall production. Agriculture 3.0 involves extensive automation. Automatic equipment improves and automates production operations.

1520-9202 © 2023 IEEE
Digital Object Identifier 10.1109/MITP.2022.3205707
Date of current version 11 January 2023.

Agriculture 4.0 is the current global goal. It is characterized by the use of agricultural knowledge of the past in cultivation activities. AI, Big Data, IoTs, and other technologies are used to collect information about crop growth status and human cultivation behaviors. Crop growth status is transmitted from a sensor device to a system and artificial intelligence (AI) is used to make cultivation decisions; crop cultivation methods are adjusted in a timely manner in response to external environmental changes. The cultivation of crops becomes both more automatic and smarter.[2]

This study is based on Agriculture 4.0, and involves the integration of IoTs, Big Data, automatic control, and AI to construct an automated and intelligent cultivation system, which allows farmers in real time to determine the status of crop and to actively perceive and learn about farmer cultivation behavior; the system learns farmers' cultivation methods and eliminates poor results that might otherwise be achieved when farmers have insufficient farming experience.

CONTRIBUTIONS

The system is developed using image recognition, AI, data analysis, and other technologies. The system assists farmers in decision-making and includes a controller to support intelligence and automation. The contributions of this work are as follows.

1) A LoRa-based IoT information collection platform is constructed to provide users with farm information behavior analysis and decision support. To overcome limits on data transmission that would otherwise be imposed by power consumption, LoRa is used for long-distance data transmission. Furthermore, a dynamic transmission mechanism is used to reduce the amount of data transmitted, reducing energy consumption.
2) Image analysis is used to identify the growth status of crops. Cultivation methods vary among the periods of a plant's growth. In this study, crop growth status is identified by a color analysis of images.
3) An automated cultivation system that irrigates according to system settings is developed. The automated cultivation system can reduce time spent in cultivation and determine more accurately the optimal amount of water that is used for irrigation. IoT devices are used to control a solenoid valve in support of remote and automated cultivation based on the user's settings.
4) An automatic irrigation system, which can determine crop growth conditions and environmental changes, is constructed. The IoTs is used to collect information; farmers' cultivation data are used to build an AI model, and finally, water consumption in each irrigation period is determined.

BACKGROUND KNOWLEDGE

Technical Introduction

1) Intelligent Agriculture: Intelligent agriculture seeks to improve upon traditional agriculture using new technologies. Technologies such as IoT, image recognition, AI, and cloud computing, can be used to improve the convenience of agricultural cultivation and management by analyzing data and more accurately choosing agricultural cultivation methods.[3] The purpose of intelligent agriculture is to manage, control, and produce comprehensively and automatically, and to execute decisions more intelligently.[4]

2) Internet of Things: The IoTs is used in various fields and involves various types of sensing devices that are integrated and development with services for application. The architecture of the IoTs is divided into application, transmission, and sensing layers. Each layer has its associated technology for data collection, analysis, and use.[5]

3) Machine Learning: Machine learning involves the use of various algorithms to extract knowledge from complex and diverse data that have been previously collected. Machine learning is classified into supervised, unsupervised, and reinforcement.[6]

4) Expert System: An expert system exploits domain expertise by the acquisition of relevant experience, input into an expert knowledge base on a computer. A complete expert system includes a knowledge base, a database, and a reasoning mechanism. The knowledge base and database store expert knowledge and reasoning rules. The reasoning rules can be interpreted so that the expert system can derive results about prediction.[7]

5) Image Recognition: Image recognition is the use of image data to obtain useful information by analysis to identify the object in the image. An image contains much recognizable information. In the past, identification of status was performed manually.[8]

Related Work

Intelligent agriculture uses information and communication technology to improve the convenience of the agricultural management, simplify its procedures, and increase output. Relevant research includes the following.

G. Kavianand et al. integrated GSM and ARM in a control irrigation system, which monitors the pH and soil nitrogen content of soil for automatic watering and cultivation.[9] Ji-chun Zhao et al. constructed a wheat cultivation system based on agricultural knowledge. The system analyzes data about pests and diseases, and their research emphasizes the use of Big Data in agriculture.[10] Lijia Sun et al. used reinforcement learning to predict agricultural water consumption for irrigation, reducing water waste. They used a Q Learning algorithm with intensive learning to construct and train models. The results show that reinforcement learning supports irrigation control based on crop growth conditions.[11] Ban Alomar et al. used fuzzy logic to support irrigation. They used a fuzzy controller to make the overall watering system more accurate and to set the watering volume in a timely manner in response to external changes.[12]

Juan M., Núñez V. et al. used wireless sensors to collect air temperature and humidity, and soil temperature and humidity, and to record production volume. They analyzed this information to obtain information that could be used to improve cultivation.[13] K. A. Patil et al. used the IoTs to collect and analyze farmland data. The transmission layer in their system uses the IEEE 802.15.4 transmission protocol and it uses ZigBee

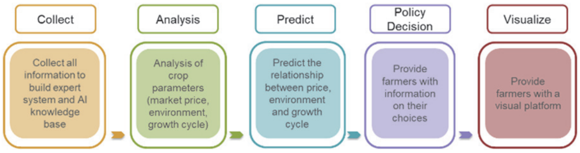

FIGURE 1. System processing flow.

to transmit data to a cloud server.[14] Wang Dongjie et al. analyzed agricultural data determine the impact of the environment on corn yield. Their methods of analysis included regression analysis, neural network and tree modeling.[15] Peng Zhang et al. constructed an irrigation system that was based on Big Data.[16]

Soumil Heble et al. used LoRa in an agricultural environment to reduce the power consumed in transmitting data.[17] Chutinan Trongtorkid et al. analyzed mango pests and diseases, and solved the problem of mango crop yield decline due to pests. They collected mango image information and expert knowledge and used a decision tree to construct a judgment model that helped in identifying mango diseases and pest insects.[18]

Erlina Agustina et al. used a rule-based expert system to solve the problem of rice pests and diseases. Their research results demonstrate that their method is effective for identifying rice pests and diseases.[19] Liu Yingying et al. constructed a correlation spanning tree with expert knowledge to diagnose and analyze tomato diseases. They collected tomato image information to identify features of diseased tomatoes and imported it into an expert knowledge base.[20]

PROPOSED AGRICULTURE SYSTEM ARCHITECTURE

This work proposes an automatic and intelligent irrigation system, which has the five functions of data collection, analysis, prediction, policy decision-making, and visualization. The collection function involves sensing the environment and collecting relevant data. The analysis function involves analyzing crop growth status and water demand for crops. The prediction function predicts irrigation water demand for the entire farm. The policy decision-making function is used for cultivating and watering crops. The visualization function is responsible for visually representing relevant information. The system processing flow is shown in Figure 1.

1) Collection: The collection function involves the use of sensor devices to collect data about the environment on the farm. Each sensing device is placed outside, so it must be stable and adaptable.

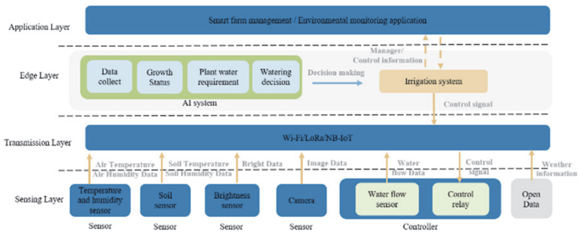

FIGURE 2. System architecture.

2) Analysis: The analysis function determines the growth status of crops. The water absorption rate of plants varies among periods, so the amount of water also varies. Images are used to determine the growth status of crops for the purpose of decision-making.
3) Prediction: The prediction function uses the crop growth status and environmental sensing information to predict the amount of water that is required for crop growth.
4) Policy Decision: The policy-decision function is based on an analysis of crop water demand and the schedule of grower; a control signal concerning crop water demand is transmitted to the controller, supporting the automated cultivation of crops.
5) Visualize: The visualization function provides a visual presentation based on collected information, the results of analysis and decision-making information. Users can thus quickly grasp current situation.

System Introduction

AI and IoT technology are used to construct an Artificial Intelligence of Things (AIoT) cultivation system. The system architecture is divided into four layers sensing, transmission, edge, and application layers, as shown in Figure 2.

1) Sensor Layer: Each sensor is connected to the irrigation controller in the sensing layer. The information that is collected in the sensing layer is transmitted to the communication layer. A photographic lens is set up outside to capture images, and the information thus collected is transmitted to a web server for storage. The collected data include air temperature and humidity, brightness, soil temperature and humidity, and crop images. Forecasts of the weather, humidity, and probability of rain are also collected and stored data in a database.

2) Transport Layer: The transport layer uses LoRa and Wi-Fi, which consume little power, to transmit data. The usefulness of a sensing device is limited by the environment, its power supply and its distance from outside so low-power LoRa communication is suitable for collecting data, and can be deployed over a wide range of farm.

3) Edge layer: An AI and irrigation module is deployed in the computing layer. The AI module analyzes plant growth status and predicts the demand for water for the crops. The irrigation module controls irrigation based on the results of decisions made in the application layer.

4) Application Layer: The application layer includes an environmental monitoring and cultivation management module. Field devices are controlled using the management module. Information is collected on the statuses of devices and environmental changes of the operator, who can operate and manage devices when an emergency occurs.

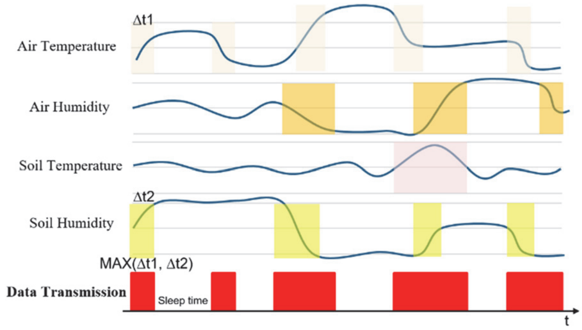

FIGURE 3. Dynamically communication mechanism.

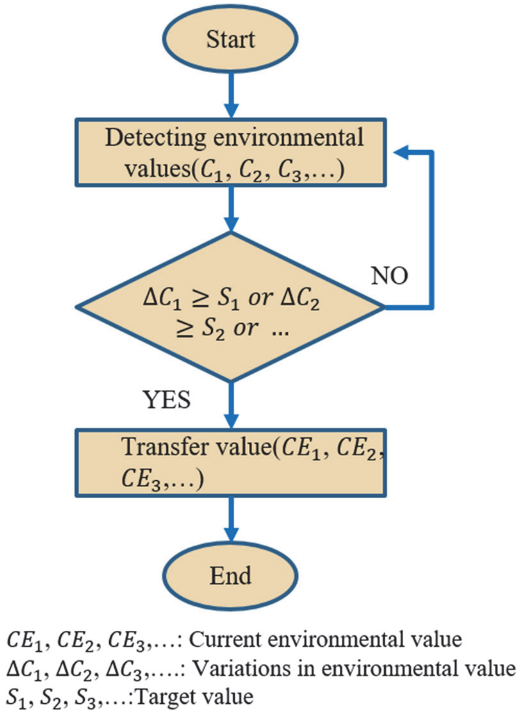

FIGURE 4. Dynamic environment detection mechanism process.

Growth Status, Plant Water Requirement, and Watering Decision Operations Function

In this work, AIOT system architecture is used in data collection, data analysis, prediction, and decision-making in intelligent agriculture. Growth Status, Plant Water Requirement, and Watering Decision operator (G.P. & W) functions are used to analyze crop growth, identify growth status, and predict water demand. Finally, decisions regarding watering are made based on predictions about farm and a farming schedule, with control signals sent to the irrigation system.

1) Data Collect: A dynamic communication mechanism that detects environmental changes was designed to save power in transmitting sensor data because outdoor IoT devices have limited power, and frequent communication can waste considerable energy. When a change in an environmental variable exceeds a threshold value, the communication device is awakened to transmit a message. When the change in the variable is less than a threshold value, the device remains in sleep mode. Accordingly, the energy that is consumed in communication is reduced. Figure 3 shows dynamically communication mechanism.

When a processor receives a detection signal, it calculates the change between the sensed value and the previous value. If the change exceeds a threshold, the communication module is awakened to transmit data. The flowchart as shown in Figure 4.

2) Growth Status Analyze: Gaussian blurring of images that are captured by cameras is used to filter out excessively detailed information and identify important features. Images are decomposed into RGB primary colors. The color ratio of the image is calculated to obtain the color of crops in various growth stages. Figure 5 shows the crop growth analysis schematic.

3) Plant Water Requirement Predict: The forecasting of water demand for crops is divided into two stages, offline and online, as shown in Figure 6. The offline stage involves data preprocessing and model establishment. The online stage forecasts the water demand for irrigation.

ANOVA is used to evaluate the importance of features in the collected data. The average of the collected data is calculated to determine the weather

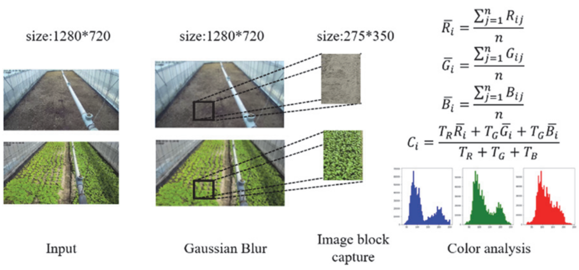

FIGURE 5. Growth status analyze.

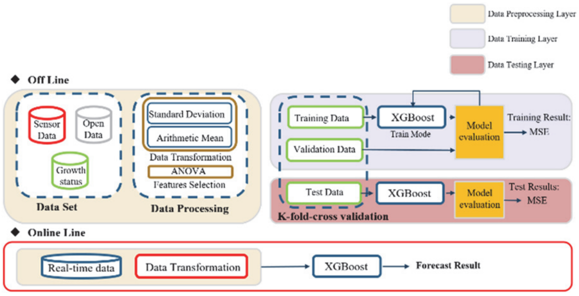

FIGURE 6. Machine learning architecture.

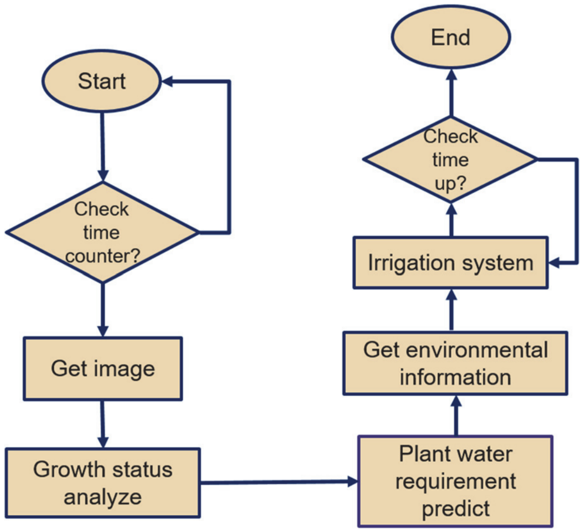

FIGURE 7. Watering decision flowchart.

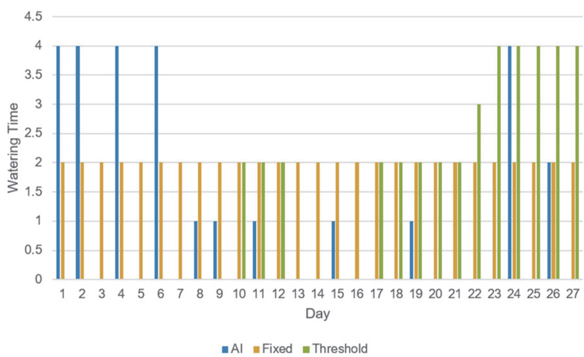

FIGURE 8. Watering time comparison (per day).

status for each day. XGBoost is used to obtain the correlation between feature values and target values, and regression prediction is used to build a model of prediction. The regression prediction is expressed by the formula (1), y_i is the predicted value, and $\hat{y}_i$ is the actual value. $L(\theta)$ represents the training error

$$\mathrm{L}\ (\theta) = \sum_i (y_i - \hat{y}_i)^2. \quad (1)$$

4) Watering Decision: After crop growth status is analyzed, the crop growth period is incorporated into the plant water demand prediction model, which determines the current water demand for the plants. The irrigation system receives the forecast of plant water demand and starts the pump for watering. The flowchart of irrigation decision is shown in Figure 7.

PERFORMANCE ANALYSIS

In this work, the consumption of water for farm irrigation is analyzed using proposed, fixed, and threshold methods. The fixed method is to water crops for two minutes a day for the entire vegetable period. The threshold method is to determine whether humidity is less than 50% and, if so, to water. Figure 8 shows the comparison of watering time per day.

Figure 9 is a comparison of cumulative watering time. The fixed method involves a total of 54 minutes of watering. The threshold method involves a total of 39 minutes of watering. The proposed method involves a total of 27 minutes of watering, and so provides the lowest water consumption.

Farms require daily soil cultivation, weeding, irrigation, fertilizer supplementation, fertility observation, plant disease prevention and control, equipment maintenance, and water tower cleaning. An AI system reduces the time required to observe the results of cultivation and determine the amount of watering required. Information transformation by the IoTs enables the overall management of data, enabling farmers to determine status of their farms more quickly. Figure 10 shows that the proposed method can reduce time by about 40.94% in process of reachability management.

Figure 11 shows a power consumption comparison. The number of timed transmissions of message

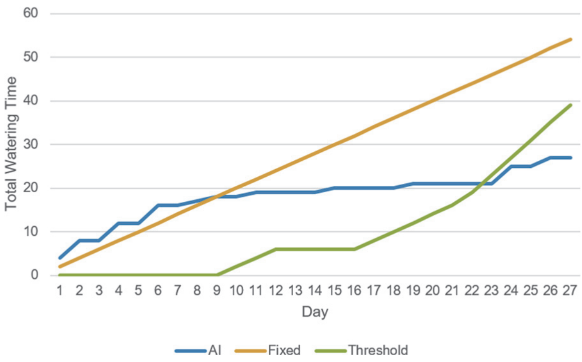

FIGURE 9. Watering time comparison (cumulative time).

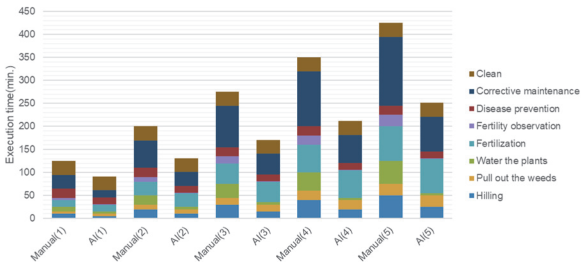

FIGURE 10. Execution time comparison.

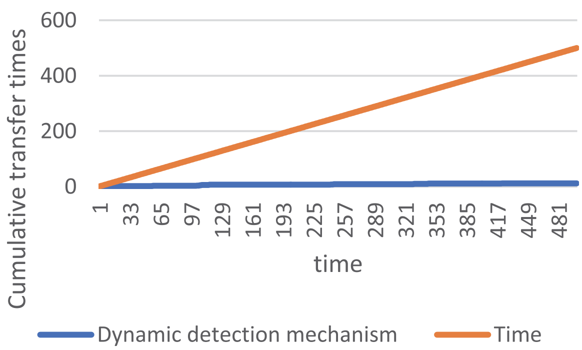

FIGURE 11. Energy consumption comparison.

delivery daily can reach 500, but the dynamic detection transmission mechanism requires only approximately 12 transmissions. The dynamic transmission reduces the number of transmissions by 97.8%.

CONCLUSION

In this work, the IoTs and machine learning are used to build a system that can automatically and intelligently cultivate crops. This system includes multiple outdoor sensing modules to collect information; uses low-power transmission LoRa to transmit information, and schedules communications to reduce the total energy consumed for communication. The purpose is to cultivate high-quality crops with minimal manpower. The proposed method effectively reduces the amount of water resources used by 50% and 30% below those required by the fixed method and the threshold method. It reduces the time that farmers must spend to observe the environment and watering operations during cultivation by 40.94%. In order to make the decision of this system closer to the considerations and observations of agricultural experts. The limitations of the currently proposed system are the need for more types of data collection and more diverse crop samples.

REFERENCES

1. X. Chen and H. Fang, "The analysis of agricultural products consumers' purchase behavior under the background of Big Data," in *Proc. Int. Symp. Comput., Consum. Control*, 2018, pp. 420–423.
2. C. Li, Y. Tang, M. Wang, and X. Zhao, "Agricultural machinery information collection and operation based on data platform," in *Proc. IEEE Int. Conf. Saf. Produce Informatization*, 2018, pp. 472–475.
3. D. Lin, C. K. M. Lee, and W. C. Tai, "Application of interpretive structural modelling for analyzing the factors of IoT adoption on supply chains in the Chinese agricultural industry," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage.*, 2017, pp. 1347–1351.
4. K. Wongpatikaseree, P. Kanka, and A. Ratikan, "Developing smart farm and traceability system for agricultural products using IoT technology," in *Proc. IEEE/ACIS 17th Int. Conf. Comput. Inf. Sci.*, 2018, pp. 180–184.
5. M. S. T. Haque, K. A. Rouf, Z. A. Khan, A. Emran, and M. S. Rahman, "Design and implementation of an IoT based automated agricultural monitoring and control system," in *Proc. Int. Conf. Robot., Elect. Signal Process. Techn.*, 2019, pp. 13–16.
6. S. A. Z. Rahman, K. C. Mitra, and S. M. Mohidul Islam, "Soil classification using machine learning methods and crop suggestion based on soil series," in *Proc. 21st Int. Conf. Comput. Inf. Technol.*, 2018, pp. 1–4.
7. G. D. Santika, D. A. R. Wulandari, and F. Dewi, "Quality assessment level of quality of cocoa beans export quality using hybrid adaptive neuro - Fuzzy Inference system (ANFIS) and genetic algorithm," in *Proc. Int. Conf. Elect. Eng. Comput. Sci.*, 2018, pp. 195–200.

8. P. Soni and R. Chahar, "A segmentation improved robust PNN model for disease identification in different leaf images," in *Proc. IEEE 1st Int. Conf. Power Electron., Intell. Control Energy Syst.*, 2016, pp. 1–5.
9. G. Kavianand, V. M. Nivas, and R. Kiruthika, "Smart drip irrigation system for sustainable agriculture," in *Proc. IEEE Technol. Innov. ICT Agric. Rural Dev.*, 2016, pp. 19–22.
10. J. Zhao and J. Guo, "Big data analysis technology application in agricultural intelligence decision system," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Big Data Anal.*, 2018, pp. 209–212.
11. L. Sun, Y. Yang, J. Hu, D. Porter, T. Marek, and C. Hillyer, "Reinforcement learning control for water-efficient agricultural irrigation," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl. IEEE Int. Conf. Ubiquitous Comput. Commun.*, 2017, pp. 1334–1341.
12. B. Alomar and A. Alazzam, "A smart irrigation system using IoT and fuzzy logic controller," in *Proc. 5th HCT Inf. Technol. Trends*, 2018, pp. 175–179.
13. M. Juan, V. Núñez, R. F. Fonthal, and M. Q. L. Yasmín, "Design and implementation of WSN for precision agriculture in white cabbage crops," in *Proc. IEEE 24th Int. Conf. Electron., Elect. Eng. Comput.*, 2017, pp. 1–4.
14. K. A. Patil and N. R. Kale, "A model for smart agriculture using ioT," in *Proc. Int. Conf. Glob. Trends Signal Processings Inf. Comput. Commun.*, 2016, pp. 543–545.
15. W. Dongjie, L. Zhemin, and W. Shengwei, "Exploring the relationship between maize yield and climate Big Data on maize belt of northeast China," in *Proc. IEEE 2nd Int. Conf. Big Data Anal.*, 2017, pp. 513–516.
16. P. Zhang, Q. Zhang, and F. Liu, "The construction of the integration of water and fertilizer smart water saving irrigation system based on Big Data," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. IEEE Int. Conf. Embedded Ubiquitous Comput.*, 2017, pp. 392–397.
17. S. Heble, A. Kumar, and K. V. V. Prasad, "A low power IoT network for smart agriculture," in *Proc. IEEE 4th World Forum Internet Things*, 2018, pp. 609–614.
18. C. Trongtorkid and P. Pramokchon, "Expert system for diagnosis mango diseases using leaf symptoms analysis," in *Proc. Int. Conf. Digit. Arts, Media Technol.*, 2018, pp. 59–64.
19. E. Agustina, I. Pratomo, and A. D. Wibawa, "Expert system for diagnosis pests and diseases of the rice plant using forward chaining and certainty factor method," in *Proc. Int. Seminar Intell. Technol. Appl.*, 2017, pp. 266–270.
20. Y. Liu, X. Zhao, and X. Zhu, "Knowledge expression and reasoning model for tomato disease diagnosis," in *Proc. 3rd Int. Conf. Inf. Manage.*, 2017, pp. 284–288.

YI-WEI MA is an assistant professor at National Taiwan University of Science and Technology, Taipei, 106, Taiwan. His research interests include Internet of Things, cloud computing, future network, and ubiquitous computing. He is the corresponding author of this article. Contact him at yiweimaa@gmail.com.

JIANN-LIANG CHEN was born in Taiwan on December 15, 1963. Since August 1997, he has been with the Department of Computer Science and Information Engineering of National Dong Hwa University, where he is a Professor and Vice Dean of Science and Engineering College. He joined the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, 106, Taiwan, as a Distinguished Professor and is currently a Dean. His current research interests are directed at cellular mobility management, cybersecurity, personal communication systems, and Internet of Things. Chen received the Ph.D. degree in electrical engineering from National Taiwan University. Contact him at lchen@mail.ntust.edu.tw.

CHING-CHIU SHIH is with the National Taiwan University of Science and Technology, Taipei, 106, Taiwan. His research interests include Internet of Things, and intelligent agriculture. Shih received the M.S. degree in electrical engineering at National Taiwan University of Science and Technology. Contact him at a9605340@gmail.com.

WASHINGTON, D.C.:
2001 L St., Ste. 700,
Washington, D.C. 20036-4928
Phone: +1 202 371 0101
Fax: +1 202 728 9614
Email: help@computer.org

LOS ALAMITOS:
10662 Los Vaqueros Cir.,
Los Alamitos, CA 90720
Phone: +1 714 821 8380
Email: help@computer.org

EDITOR: Theresa-Marie Rhyne, theresamarierhyne@gmail.com

DEPARTMENT: VISUALIZATION VIEWPOINTS

This article originally appeared in IEEE Computer Graphics and Applications vol. 44, no. 2, 2024

Databiting: Lightweight, Transient, and Insight Rich Exploration of Personal Data

Bradley Rey, University of British Columbia, Kelowna, BC, V1V 1V7, Canada
Bongshin Lee, Microsoft Research, Redmond, WA, 98052, USA
Eun Kyoung Choe, University of Maryland, College Park, MD, 20742, USA
Pourang Irani, University of British Columbia, Kelowna, BC, V1V 1V7, Canada

As mobile and wearable devices are becoming increasingly powerful, access to personal data is within reach anytime and anywhere. Currently, methods of data exploration while on-the-go and in-situ are, however, often limited to glanceable and micro visualizations, which provide narrow insight. In this article, we introduce the notion of databiting, *the act of interacting with personal data to obtain richer insight through lightweight and transient exploration. We focus our discussion on conceptualizing databiting and arguing its potential values. We then discuss five research considerations that we deem important for enabling databiting: contextual factors, interaction modalities, the relationship between databiting and other forms of exploration, personalization, and evaluation challenges. We envision this line of work in databiting could enable people to easily gain meaningful personal insight from their data anytime and anywhere.*

As device hardware and software advance, enabling broader access to personal data, new opportunities for mobile data exploration arise: mobile data exploration has the potential to intertwine with our lived experiences and day-to-day activities. However, conducting data exploration in on-the-go scenarios of use poses unique challenges. It is crucial that exploration does not hinder, but rather assists, a wide range of scenarios and contexts in which we find ourselves seeking insight.[1] In this article, we advocate for the visualization and personal informatics research communities to focus on the development of lightweight and transient exploration techniques that enable insight rich access to personal data.

Current systems offer one of two approaches to mobile data exploration. Glanceable and micro visualizations have been widely adopted in mobile applications,[2] at times combined into dashboards (Figure 1 left). They provide concise and focused representations of information in a limited space and context for users to easily grasp information at a glance. However, despite their popularity, they offer only specific insights and allow limited interaction, leaving users without the ability to cater to their personal and situational needs. In contrast, heavyweight applications have been designed to enable more comprehensive data exploration (Figure 1 right). These applications often require considerable time and knowledge to use. These barriers make them at times inaccessible or inconvenient (e.g., during physical activity, while walking a pet, or while cooking). Between these approaches, a significant gap in the field of personal informatics and visualization arises: limited information richness hinders users' ability to better comprehend and leverage personal data through exploration that can be efficiently undertaken during broader contexts.

We discuss the notion of *databiting*, a term we coined to indicate lightweight, transient, and insight

0272-1716 © 2024 IEEE
Digital Object Identifier 10.1109/MCG.2024.3353888
Date of current version 25 March 2024.

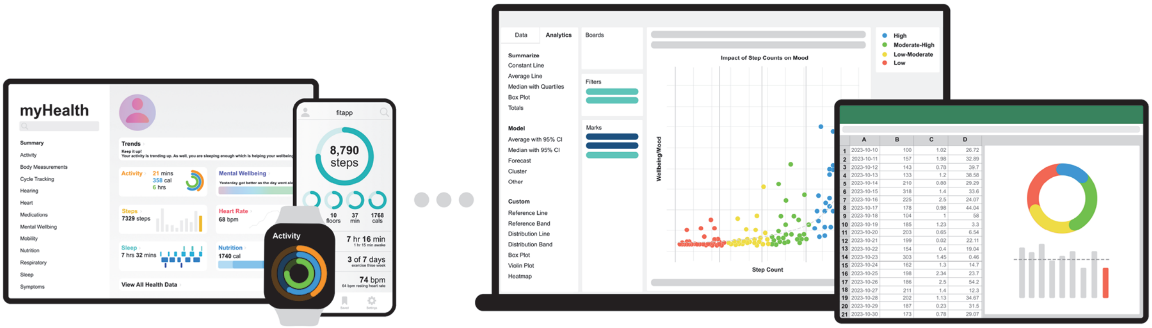

FIGURE 1. Representations of current mobile data exploration applications are highlighted, grouped by general information richness: applications which afford glanceable and micro visualizations (at times combined into dashboards) (left) and applications, which provide potential for heavyweight data analysis (right).

rich exploration. Achieving higher levels of information richness often proves challenging. However, we should actively seek ways to enhance accessibility and ease in obtaining greater insights. By embracing the approach outlined in this article, we can empower individuals to effortlessly gain insights from their data as needed, transforming the way they explore and interact with personal data on-the-go. Together, this recognizes the importance of our interactions with personal data and emphasizes the significance of seamlessly integrating rich personal insights into our daily lives.

To better conceptualize databiting, we provide a storied real-life example (Figure 2). While this example highlights the use of a smartwatch, databiting can be beneficial for any device on which personal data can be explored. Imagine Sam, while hiking, checks their smartwatch and becomes aware of a possible shortfall in their exercise. The smartwatch initially displays only current metrics. Seeking a deeper understanding, Sam taps and holds on the pace number, and adds a natural language query, stating *"Compare to my last 6 hikes."* In response, a graph appears indicating a relatively slower pace for the day. With this insight, Sam decides to increase their pace for the remainder of the hike, determined to reach their average by the end of the trail. This seamless exploration of personal data during an activity is made possible through databiting, during which individuals can quickly and easily gain meaningful insight into their data on-the-go.

In this article, we introduce the concept of databiting and argue for its potential value. We then delineate five research considerations—contextual factors, interaction modalities, the complementary relationship between databiting and other forms of exploration, personalization, and evaluation challenges—focused

FIGURE 2. To the left of the red dashed line is one example of the current state of exploration for mobile data visualizations. To the right, databiting is enabled for physical activity related information. This has the potential to provide greater influence and insight of the current activity taking place. Notably, more in-depth long-term data exploration is left for a later time.

toward enabling and understanding databiting. Importantly, these research areas can work in concert to provide lightweight and transient access to richer personal insight anytime and anywhere.

PERSONAL DATA EXPLORATION

Personal informatics, an interdisciplinary field encompassing human–computer interaction and visualization research, emphasizes the collection, comprehension, and utilization of personal data.[3] At its core, personal informatics aims to leverage data for enhanced personal insights and actionable knowledge. Achieving this goal involves a multifaceted approach that entails understanding people's unique needs and patterns of usage, as well as designing and evaluating innovative technologies, interaction modalities, and data representations.

Thanks to the increasing capability and prevalence of mobile and wearable devices, the collection of and access to personal data is quickly growing. From this data, people have the potential to gain insights into their behaviors and various aspects of their lives, such as health, finances, and social connections. While collecting and presenting personal data is essential, true value emerges when individuals can explore and interact with their data in meaningful ways.

From glanceable and micro visualizations to interactive dashboards on mobile and desktop devices, the aim is to create interfaces that empower individuals to effortlessly explore and gain insights from their personal data (e.g., uncovering patterns, trends, and correlations). However, current interfaces on mobile devices often lack affordances to provide richer and personalized insight. For example, dashboards only offer a limited number of set filtering options, and glanceable and micro visualizations are predefined without interaction. This inability to further explore personal data can be seen as a form of limited functionality, which is one cause of frustration and even abandonment of wearable devices.[4]

We recognize that exploration of personal data can be embedded within activities and experiences (i.e., in-situ). We must strive for innovative solutions that empower people to easily explore their data whenever they see fit, enabling them to derive meaningful insights and make informed decisions. As we establish and refine the notion of databiting, we seek to expand ways in which individuals can perform visual data exploration in a wide array of usage scenarios.

THE DATABITING CONCEPT

We conceptualize *databiting* as the act of interacting with personal data to gain increasingly rich insight through lightweight and transient exploration. The result is a *databite*, concise personal insight that extends upon what can be derived from glanceable or micro visualizations. Databiting as both a new concept and a topic for research is fluid in nature: Boundaries defining insight and data exploration methods allowing for such insight are not rigidly defined or fixed.

To illustrate this concept, we draw upon analogy. Databiting can be seen as equivalent to eating a small and easily consumable snack. The size of a snack and the number of bites required may vary from person to person and from context to context. Yet, what remains constant is the lightweight and transient nature of snacking compared to consuming a meal (often a reasonably large amount of food). In the context of data exploration, databiting equates to the consumption of bite-sized information that provides rich insights or sustenance in the moment. This builds upon simply viewing a mobile data visualization and does not require more in-depth and long-term data exploration, which can be done later when necessary or more appropriate.

Importantly, databiting is not meant to replace either in-depth exploration of data or shorter-form viewing of glanceable visualizations; rather, it is complementary to them. By bridging the gap between the two forms of exploration methods, databiting offers a new form of complementary exploration that pushes the boundaries of what is attainable. This integration of exploration methods can foster a more comprehensive, valuable, and unique (i.e., richer) understanding of personal data, yet remain accessible in a lightweight and transient manner. By offering a range of exploration options, across devices and throughout a range of usage scenarios, we expect individuals can derive greater benefits from their data-driven insights anytime and anywhere.

Lightweight, Transient, and Insight Rich Exploration

Time and effort, which should remain small, are key factors in the context of databiting. Glanceable and micro visualizations excel at providing quick and easy data engagement, however, remain limited in their level of insight conveyed.

We envision that there remains a huge opportunity to enable access to more information rich insights while maintaining a lightweight and transient approach (Figure 3). Consider a scenario in which a runner is stopped at a traffic light, waiting to cross the street. The primary task is their run and the focus on the surrounding environment. Secondary to this, they look at their heart rate zone data. Seeking a databite they simply tap on a

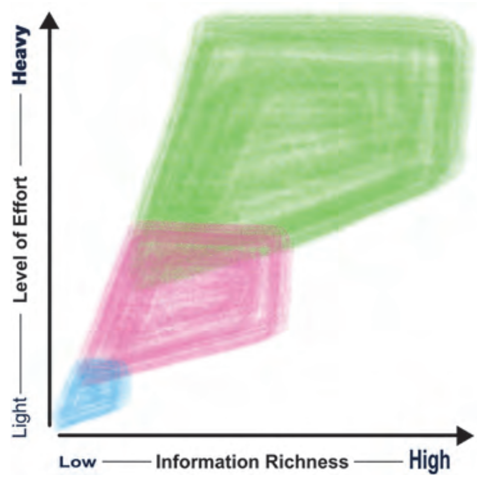

FIGURE 3. In *blue*, we highlight glanceable and micro visualizations for data exploration. In *green*, we highlight heavyweight data exploration and analysis. In *pink*, *databiting*, as a concept, promotes the need for increasing information richness while exploration remains lightweight and transient. We encourage the reader to envision how visual data exploration can consume this area of the graph. We present these rectangles as *sketched* illustrations to signify the fuzzy boundaries of these forms of exploration and potential variance within them.

stacked bar chart, which highlights their current heart rate zone. This action reveals increasingly detailed insight into the time spent in each heart rate zone, enabling the runner to concentrate on entering or maintaining a specific zone as they proceed with their run.

As can be seen, databiting can offer richer insight without requiring substantial effort, engagement, or time. This allows data exploration to occur as secondary tasks, alongside a primary ongoing activity (e.g., while out for a walk) or during a recurring daily activity (e.g., riding a bus home). Through the prioritization of simplicity rather than detail and complex insight, the small "size" of databites ensures that it can be easily and appropriately consumed.

Device Agnostic Exploration

Due to their highly portable nature, the immediate availability of data being collected, and the smaller form factors limiting heavier exploration, databiting is particularly well-suited for smartwatches and other wearable devices. However, the emphasis of databiting is not on the specific device used or insights gained but is in the manner in which data are accessed. As such, databiting can be seen as device agnostic and can be done on any device that grants access to relevant data, ranging from wearables, smartphones, and tablets, to laptops. For instance, before switching to a different task on a laptop, a researcher first quickly checks the current screen-time information. With simple mouse interactions, they learn about a negative trend that results from increasing the duration of one continuous work block and, thus, decide to take regular breaks, starting with an immediate break. Similarly, a tablet user exploring their financial data while in transit could use the stylus to circle a transaction and draw a line to another to view how they are related. Taking the interaction a step further, the user could employ predefined gestures to accomplish further exploration (e.g., displaying a transaction timeline or finding similar transactions). Multiple databites could be made throughout their journey as they look through their most recent bank statement.

Data Agnostic Exploration

The versatility of databiting extends beyond that of personal data, making it applicable to a wide range of domains and contexts involving data activities. While the initial conceptualization revolves around personal data for individual insights, the fundamental principles of lightweight, transient, and insight rich exploration can be seamlessly applied to various data categories. For instance, in the business context, professionals can benefit from quick insights (e.g., during a meeting). Databiting can enable easy access to necessary insights to foster discussions and inform decision-making without the need for extensive data preparation or analysis. In the context of scientific research, researchers running a Prolific (pilot) study can use databiting to garner quick insights about the amount and quality of collected data. This can help them decide to continue or stop and refine, if needed.

EXPECTED BENEFITS OF DATABITING

We discuss envisioned benefits of databiting that have the potential to overcome limitations and challenges of the current capabilities of mobile data exploration. Further study is needed to identify and demonstrate any tangible benefits that may exist.

Introductory and Intermediary Access

As highlighted earlier, databiting has the potential to bridge the gap between brief information access and substantial knowledge gain. Offering lightweight, transient, and insight rich access to personal data allows for engagement with smaller bite-sized snippets of timely information. This approach provides an accessible entry point for individuals new to data exploration to explore and understand their data.

Furthermore, a bite-sized approach to exploring data can serve as an intermediary step, providing individuals with a gateway for more in-depth data exploration. By offering exploration in easily consumed pieces, databiting can spark further interest, familiarity, and excitement. When presented with quick and accessible personal insights, individuals may become more curious and motivated to further explore their data. Over time, this increased engagement may foster a greater sense of familiarity and confidence with data; ultimately facilitating further exploration and a deeper understanding.

Increased In-Situ Insight

Due to the lightweight, transient, and insight rich nature, we anticipate databiting will be beneficial for obtaining data-driven insights during in-situ exploration. In-situ instances of exploration refer to moments in which data analysis and reflection occur closely related to an ongoing activity, enabling immediate and direct impact.[5] Databiting can extend in-situ data access, currently offered through glanceable and micro visualizations and dashboard manipulation, by providing further information richness. This leads to increased actionable insights that are directly relevant to the ongoing activity. Whether it is using a smartphone to explore caloric intake for the day while cooking, asking a smart speaker about your home energy consumption for the day as you arrive from work, or checking productivity on your laptop while studying, databiting can empower individuals to gain valuable personalized insights that inform immediate decision-making while in situ.

Perceived Usefulness

A current challenge concerning devices and applications that allow for personal data access and exploration is a lack of perceived usefulness.[4] Current offerings often fail to meet expectations, resulting in the abandonment of devices, applications, and even the collection of data altogether.[6] While addressing perceived usefulness is a multifaceted challenge, databiting can serve as a promising start. Increased information richness and personalized insights enabled by databiting have the potential to enhance the perceived usefulness of the devices and applications used for databiting, and the value derived from collected data. In turn, this could lead to greater overall outcomes as the collection and exploration of personal data are not outright abandoned.

RESEARCH CONSIDERATIONS

In this section, we discuss how we can begin to enable databiting through research, to achieve increased access to personal data anytime and anywhere. Specifically, we discuss research considerations that aim to overcome existing challenges and benefit from promising opportunities.

Contextual Factors

Contextual factors can become increasingly integrated with databiting, shaping both the recognition of potential exploration and the integration of relevant data. Unlike conventional data exploration, in which context might be analyzed as an additional factor later on, databiting can be seen to easily integrate context. Glanceable visualizations, being static in nature, offer limited contextual impact. At most, they present predefined visuals, such as showing a sleep chart immediately upon waking up.

Yet, due to the potential for databiting to cater to an array of in-situ scenarios, contextual information can and should be incorporated. Imagine Dani, a traveler, using a mobile app to explore a new city. Basic on-screen insights might reveal Dani's location in relation to nearby attractions. However, when databiting, Dani, and even the app itself, can leverage contextual data such as time of day, personal preferences, and location history. This integration of contextual data can enhance the overall experience and provide the potential for increasingly relevant in-situ insights.

Focusing research on incorporating contextual factors within data exploration can significantly impact databiting. For example, we can study techniques to incorporate contextual attributes into data visualizations, develop context-aware recommendation algorithms, and look to gain a better understanding of how context can be incorporated into queries desired for databiting. By embracing contextual data, databiting can further in-situ exploratory capability and insight.

Interaction Modalities

Enabling databiting requires considering appropriate interaction modalities that are not only efficient but also cater to the unique constraints of in-situ scenarios. This is not straightforward, especially because we aim to increase the level of data accessibility across devices with potentially limited interaction spaces (e.g., smartwatches, smartphones, augmented reality glasses).

To facilitate lightweight, transient, and insight rich exploration of personal data, a multimodal approach through natural language and the device's primary interaction method can be used (see Figure 2). The primary input modality (e.g., touch) allows for direct manipulation, discrete selections, and when natural language keywords cannot be remembered.[7] Importantly, natural language (e.g., speech) benefits from enabling fast and flexible expression of complex queries.[8] Recent research in personal health applications on smartphones has demonstrated the benefits of multimodal touch and speech interactions for gaining insights into personal health data.[7] Notably, individual and combined interactions were often used for differing, yet equally important, components of the data exploration process. Touch and speech combined showed promise for refinement of an initial query or displayed graph, much like databiting may extend upon glanceable or micro visualizations. However, a better understanding of individuals' personal data query requirements needs to be further addressed to fully recognize the interactive needs of databiting, and is beginning to see effort.[9]

Furthermore, exploring alternative output modalities can complement currently available data visualizations. Devices such as headphones, earbuds, and home assistants offer opportunities to incorporate natural language responses when databiting. This can appropriately provide access to data when people's visual systems are overloaded.[10] There is limited work on the requirements for natural language responses to personal data queries. Further research can focus on formulating efficient responses, determining the level of conveyed insights, and ensuring the memorability of responses. Integrating these findings with existing data visualization approaches can optimize access to personal data.

Databiting and Broader Exploration

The nature of databiting invites opportunities to consider a relationship with further in-depth data exploration. Understanding how people can transition from one to the other and how previous in-depth exploration can inform future databiting will contribute to a cohesive and personalized data exploration experience. However, it is crucial to recognize that the seamless transition between databiting and broader exploration experiences will likely require extensive device and app interoperability. Future research is needed to address these technical hurdles.

Databiting can afford a stepping stone for people to delve into in-depth data exploration when more appropriate. For example, as seen in Figure 2, Sam may recognize that their pace has been on a decline for a while. Not sure why, they can save the databite such that they are reminded to further explore for an external cause (e.g., less sleep) at a later time (e.g., while at a desk) and on an appropriate device (e.g., on a tablet). It becomes important to understand how individuals can save and manage databites, and any new questions that arise, for future exploration.

Conversely, the results of prior in-depth data exploration have the potential to influence and enrich future databiting. For example, an individual may have taken the time to pore through their credit card statements, itemizing transactions into categories and noting a budget limit for each. When in the store, this past exploration may influence databiting which is directed toward understanding if a purchase can and should be made within a predefined category.

These examples further highlight an underlying challenge concerning interoperability. Currently, fragmentation and nonstandardized access to collected data hinders cohesive personal data exploration. For instance, data saved on one device may not seamlessly integrate with a user's preferred in-depth exploration tool on another device (e.g., to analyze collected data from an Apple Watch in Tableau, exporting and importing of data is required). Furthermore, data collected on multiple devices may be stored using different protocols and formats. Without a concerted focus on interoperability, the potential for databiting to seamlessly complement broader exploration methods, and vice versa, may be hampered, limiting overall effectiveness.

Personalization

Studying and supporting personalization is crucial for optimizing databiting and enhancing the exploration process. By tailoring insights, recommendations, and visualizations to individuals and contexts of use, personalization ensures that relevant information is efficiently presented. This reduces the required time and effort, enabling people to quickly access valuable insights and make informed decisions.

Through personalized databiting, technology mediation (i.e., artificial intelligence, machine learning, etc.) can alleviate individuals from manual data analysis. For example, a personalized databite may provide instant workout recommendations. A potential reduction in temporal workload allows people to focus on the actionable outcomes of databiting rather than the exploratory process. Furthermore, serendipitous discovery is a benefit of visual data exploration. Personalization can be used to mediate tailored serendipitous insight, even with lightweight and transient exploration.

Evaluation Challenges

To realize the full potential of databiting, it is essential, yet challenging, to understand an individual's needs and goals when interacting with personal data. Modest methods such as sketching or surveys lack real-world data, while heavyweight approaches through the creation of working mobile applications are costly and time-consuming. Balancing methods through data engagement interviews and Wizard-of-Oz studies can help capture needs in various daily contexts, although in-lab study methods may suffer from recall bias.[11]

To assess the value of databiting, measuring effectiveness of personalized exploration and insight enabled by databiting is important. Longitudinal studies can capture behavior change over time, but studying and capturing the immediate influence of databiting is not straightforward. Understanding the impact of databiting compared to glanceable and micro visualizations, and longer form data exploration would also be valuable.

With the above in mind, accommodating in-situ studies is crucial for a comprehensive understanding of databiting. Current methods such as experience sampling[12] and diary studies,[13] while valuable, have limitations in capturing the full range of potential scenarios. Further developing appropriate research methods within situated environments is required. By capturing real-world usage and contextual factors, we can gain a more nuanced understanding of how individuals engage with databites in their everyday lives, further advancing the field and maximizing potential.

CONCLUSION

We conceptualize *databiting* as the process of extracting rich insight through lightweight and transient data exploration. Fueled by the widespread use of ubiquitous high-performance sensing and interactive devices, coupled with the increasing collection of personal data and situated environments of use, this form of exploration has the potential to transform the way people interact with and understand their personal data in their daily lives. By bridging the gap between glanceable and micro visualization and heavyweight visual exploration, databiting can provide complementary insight, enabling people to better access their personal data.

In this article, we have introduced databiting, provided examples and expected benefits, and delineated research considerations that remain to be undertaken. We envision that this key, yet underexplored, concept of personal informatics may soon become a reality. We hope this work inspires research communities toward the creation of applications and tools that enable databiting. We anticipate that both exciting challenges and opportunities will arise, which in turn will shape the future of databiting, and access to personal data anytime and anywhere, for the better.

ACKNOWLEDGMENTS

The work of Eun Kyoung Choe was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (RS-2023-00263102). The work of Pourang Irani was supported in part by an NSERC Discovery Grant on In-Situ User Interfaces, from which Bradley Rey was partly funded.

REFERENCES

1. B. Lee, E. K. Choe, P. Isenberg, K. Marriott, and J. Stasko, "Reaching broader audiences with data visualization," *IEEE Comput. Graphics Appl.*, vol. 40, no. 2, pp. 82–90, Mar./Apr. 2020, doi: 10.1109/MCG.2020.2968244.
2. T. Blascheck, F. Bentley, E. K. Choe, T. Horak, and P. Isenberg, "Characterizing glanceable visualizations: From perception to behavior change," in *Proc. Mobile Data Vis.*, 2021, pp. 151–176, doi: 10.1201/9781003090823-5.
3. I. Li, A. Dey, and J. Forlizzi, "A stage-based model of personal informatics systems," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2010, pp. 557–566, doi: 10.1145/1753326.1753409.
4. S. Mujahid, G. Sierra, R. Abdalkareem, E. Shihab, and W. Shang, "An empirical study of android wear user complaints," *Empirical Softw. Eng.*, vol. 23, pp. 3476–3502, 2018, doi: 10.1007/s10664-018-9615-8.
5. I. Li, A. K. Dey, and J. Forlizzi, "Understanding my data, myself: Supporting self-reflection with ubicomp technologies," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 405–414, doi: 10.1145/2030112.2030166.
6. J. Clawson, J. A. Pater, A. D. Miller, E. D. Mynatt, and L. Mamykina, "No longer wearing: Investigating the abandonment of personal health-tracking technologies on craigslist," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 647–658, doi: 10.1145/2750858.2807554.
7. Y.-H. Kim, B. Lee, A. Srinivasan, and E. K. Choe, "Data@Hand: Fostering visual exploration of personal data on smartphones leveraging speech and touch interaction," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, pp. 1–17, doi: 10.1145/3411764.3445421.
8. J. Aurisano et al., "Show me data": Observational study of a conversational interface in visual data exploration," in *IEEE VIS*, 2015, pp. 1–2, doi: 10.13140/RG.2.2.34456.90889.

9. B. Rey, B. Lee, E. K. Choe, and P. Irani, "Investigating in-situ personal health data queries on smartwatches," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 4, pp. 1–19, Jan. 2023, doi: 10.1145/3569481.
10. S. Brewster, J. Lumsden, M. Bell, M. Hall, and S. Tasker, "Multimodal 'eyes-free' interaction techniques for wearable devices," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2003, pp. 473–480, doi: 10.1145/642611.642694.
11. G. M. Harari, N. D. Lane, R. Wang, B. S. Crosier, A. T. Campbell, and S. D. Gosling, "Using smartphones to collect behavioral data in psychological science: Opportunities, practical considerations, and challenges," *Perspectives Psychol. Sci.*, vol. 11, no. 6, pp. 838–854, 2016, pMID: 27899727, doi: 10.1177/1745691616650285.
12. S. Consolvo and M. Walker, "Using the experience sampling method to evaluate ubicomp applications," *IEEE Pervasive Comput.*, vol. 2, no. 2, pp. 24–31, Apr./Jun. 2003, doi: 10.1109/MPRV.2003.1203750.
13. N. Bolger, A. Davis, and E. Rafaeli, "Diary methods: Capturing life as it is lived," *Annu. Rev. Psychol.*, vol. 54, no. 1, pp. 579–616, 2003, doi: 10.1146/annurev.psych.54.101601.145030.

BRADLEY REY is currently a Ph.D. candidate with the University of British Columbia, Kelowna, BC, V1V 1V7, Canada. He is the corresponding author of this article. Contact him at reyb@student.ubc.ca.

BONGSHIN LEE is a senior principal researcher with Microsoft Research, Redmond, WA, 98052, USA. Contact her at bongshin@microsoft.com.

EUN KYOUNG CHOE is an associate professor with the College of Information Studies, University of Maryland, College Park, MD, 20742, USA. Contact her at choe@umd.edu.

POURANG IRANI is a professor with the University of British Columbia, Kelowna, BC, V1V 1V7, Canada. Contact him at pourang.irani@ubc.ca.

Contact department editor Theresa-Marie Rhyne at theresamarierhyne@gmail.com.

EDITOR: Mark Campbell, EVOTEK, mark@evotek.com

DEPARTMENT: IT INNOVATION

This article originally appeared in
Computer
vol. 56, no. 2, 2023

Tomorrow's Applications Require IT Operations That Are Autonomous, Ubiquitous, and Smarter—In a Word, Invisible

Mark Campbell, EVOTEK

The velocity and volume of evolving enterprise workloads are reshaping IT infrastructure and require IT operations to become increasingly invisible.

As today's businesses become more distributed, require higher velocity, and continuously change application and data ecosystems, the underlying IT infrastructure is experiencing seismic shifts. As IT infrastructures become more dispersed, diversified, intelligent, and autonomous, IT operations must be reinvented.

TODAY'S IT INFRASTRUCTURE

Today's IT infrastructure is undergoing a transition from a static, data center-bound, handcrafted, and siloed structure to a hybrid, fluid, automated, and stratified resource mesh. This transition is fundamentally changing several areas of IT Infrastructure:

- *Computational infrastructure (compute)*[1]: Today's compute platforms, delivered by physical "bare metal" servers and abstracted in virtual and microservice stacks, are no longer built to support a single application workflow. Instead, these platforms are increasingly deployed as a capacity pool allocated for various application stacks. These computational pools are built on a variety of compute platforms beyond the general-purpose CPU, including chips designed for graphics, inference, modeling, analytics, blockchain, memory-centric computation, and high-performance computing. Applications are increasingly being written to take advantage of specific chipsets and segmented to run specific functions on appropriately specialized compute cores. In addition, today's on-premises compute environments can "cloudburst" with nearly limitless resources for traffic fluctuations and unexpected outages.
- *Storage*: As today's applications evolve, underlying storage housing associated application information is transitioning from monolithic arrays of block, file, and object data into a sophisticated, adaptable, and rugged storage fabric distributed across data center location and cloud providers. These systems automatically float data between fast and expensive memory-centric media and slower but cheaper devices. Modern storage systems use advanced features, such as deduplication, compression, encryption, replication, and archiving, to provide resilient data storage in the smallest footprint possible. Many data fabrics implement a global name space to virtualize dataset storage location so that applications need not know where the data are physically stored to retrieve and modify them. This allows data to dynamically move among storage arrays, data centers, and even clouds transparently.[2]
- *Network*: Connecting ever-expanding and evolving compute and storage assets across

Digital Object Identifier 10.1109/MC.2022.3227656
Date of current version: 8 February 2023

on-premises, cloud, and edge infrastructure is a daunting task. Today's network infrastructure is too large, fast, and complicated to be managed by humans alone. This has led to more automated approaches, such as intent-based networking, software-defined networking, and software-defined wide area networking in which operations teams express their desire and systems determine network asset deployment and reconfiguration to meet these intents. Once deployed, these policies adapt the network to environmental changes, such as volume fluctuations and segment outages.[3] "Today's networks are becoming more adaptive, reliable, secure, and automated," notes Suresh Katukam, cofounder and chief product officer at Nile, "with deep instrumentation and 360-degree monitoring of all elements to feed analytics, baselining, and trending tools."[4]

- *Data center*: Whether enterprise-owned or leased from a colocation site, data centers continue to house IT infrastructure for many enterprises. In recent years, data centers have been distributed to support "follow-the-sun" operations, provide geographic redundancy, and meet regional compliance restrictions. Other enterprises are creating smaller micro- and even nano-data centers to put infrastructure closer to consumption points.
- *Cloud*: Nearly all enterprises today use infrastructure housed with public cloud providers. Most have developed a multicloud strategy to use the right cloud provider to provide specific infrastructure requirements. While there are a few cloud-only enterprises, most companies rely on a hybrid cloud architecture spanning data centers and cloud resources.
- *Edge*: By 2025, more data will be created outside of the core (that is, data centers and cloud infrastructure) than in it. These enormous data volumes are exceedingly difficult to move due to physical latency and throughput issues that make it easier to move compute to data rather than the reverse. This phenomenon, known as data gravity, is pulling applications and the supporting infrastructure outside the controlled walls of cloud and data centers into the amebic world of sensors, devices, endpoints, and mobile platforms.[5]

TODAY'S IT OPERATIONS

Today's IT operations are governed by various methodologies that control how IT organizations deploy, operate, support, and optimize IT infrastructure and services. These IT services are typically built on a deploy–measure–response lifecycle with humans in the loop controlling and verifying each stage.

TODAY'S IT INFRASTRUCTURE IS UNDERGOING A TRANSITION FROM A STATIC, DATA CENTER-BOUND, HANDCRAFTED, AND SILOED STRUCTURE TO A HYBRID, FLUID, AUTOMATED, AND STRATIFIED RESOURCE MESH.

IT operations are conducted through a suite of defined services to manage events, problems, releases, capacity, performance, change, configuration, service levels, service requests, continuity, and infrastructure deployment via a structured service desk and service catalog. These IT services are commonly defined and controlled by IT operations management and IT service management frameworks, such as Control Objectives for Information and Related Technology, the Enhanced Telecom Operations Map, the Information Technology

Infrastructure Library, the Microsoft Operations Framework, the Scaled Agile Framework, Six Sigma, and The Open Group Architecture Framework. These services are implemented by various on-premises, software as a service-based, and hybrid tools.[6,7]

With the rapid growth of cloud-based infrastructure, many operational workflows are integrated with public cloud providers' operations platforms, such as Amazon Web Services Cloud Operations, Microsoft's Azure Operations Management Suite, and Google Cloud's Operations Suite. These platforms allow hybrid cloud workflows by extending IT operations services across both data center and cloud-based operations.[8]

INFRASTRUCTURE WILL BECOME DISAGGREGATED FROM THE WORKFLOWS ITS SUPPORTS AND, INSTEAD, BE PRESENTED AS COMPUTE, STORAGE, AND NETWORK CAPACITY POOLS.

Infrastructure deployment automation is rapidly becoming a mainstay for IT operations teams and provides automated deployment and configuration of compute, network, and storage resources. Technologies such as infrastructure as code, IT automation and orchestration, DevOps, and continuous integration/continuous deployment are enabling enterprises to stand up, tear down, and reconfigure IT infrastructure and application workflows on the fly.[9]

Over the past several years, artificial intelligence-based IT operations (AIOps) have emerged as a promising development in automated incident detection and remediation. AIOps use analytical models to scan log entries, device telemetry, and/or network packet captures to learn a system, network, or enterprise's "normal" operational baseline and then spot any deviations. These anomalies are then reported to operators for remediation and, increasingly, used to trigger automated remediation scripts without operator intervention and oversight. AIOps tools also correlate various alarms, alerts, and uncharacteristic system behavior to determine root causes to aid manual and automated remediation.[10]

TOMORROW'S INFRASTRUCTURE

As tomorrow's IT infrastructure continues its current trajectory to become more multicloud, hybrid cloud, and automated, emerging patterns will prevail:

- *Predictive infrastructure*: As embedded AI models become more prevalent in IT devices, the underlying infrastructure will become more predictive. Systems will no longer just report when a failure occurs or a capacity threshold is reached. Instead, they will predict when to initiate preventive maintenance, adjust configurations, and even execute a calculated course of action. "We should think of intelligent infrastructure more like an electric grid that is simple, reliable, and secure—always on and invisible. That's a better definition," observes Katukam.[4] Systems will also predict upcoming capacity issues and suggest preemptive expansion, reconfiguration, and workload migration before system limits are hit. Beyond this, systems will increasingly not just report remediation recommendations but automatically undertake preventive measures themselves and report effected changes back to operational monitoring systems.[11]
- *Edge superstructure*: Two trends will transform the edge as we know it. First, we will see the morphing of the Internet of Things, autonomous vehicles, smart devices, and endpoint devices from distinct edge infrastructure architectures into a heterogeneous edge superstructure. Second, the center of data gravity will shift from core resources to the edge superstructure and pull workloads out of the core. Edge compute, network, and storage platforms will not only proliferate but deliver higher capacity, lower power consumption, increased performance, and more independent operations without core system oversight. "To combat data gravity, tomorrow's edge storage will rely more on metadata than the underlying data itself, will cache 'hot' data to reduce trips to the core systems and will stage data replication to the core for eventual consistency," predicts Gou Rao, cofounder and chief technology officer of Portworx.[12] We will also see a stratification between edge and

core compute systems through the evolution of "fog computing," which will place micro- and nano-data centers (or, conversely, macro- and mega-edge systems, depending on your point of view) closer to consumption points.[13]

- *Grid-based data centers*: As data center architecture evolves to follow the microservice model of hyperscaler cloud providers, we will see less custom data center infrastructure built for a specific system or workflow. Infrastructure will become disaggregated from the workflows its supports and, instead, be presented as compute, storage, and network capacity pools allocated for required workflow demands. While grid computing has been an architectural objective for several decades, recent advancements in composable infrastructure and control software are making this a reality. As legacy systems are retired, grid-based disaggregated infrastructure will become the norm.[14]

TOMORROW'S OPERATIONS

The changes in infrastructure will have disruptive impacts on tomorrow's IT operations, including

- *Ubiquitous operations*: Operations will no longer be siloed into cloud, data center, and edge frameworks. Operational workflows and services will span all environments and be able to migrate capacity and workloads accordingly as demands dictate.
- *Smart operations*: Operations will no longer just react to arising incidents but forecast failures proactively. Predictive models will also allow operational services to migrate workloads across environments before anticipated outages and capacity shifts. Operational services will also be cognizant of the cost and performance characteristics of the underlying infrastructure to optimize between these two opposing goals, depending on changing enterprise objectives.
- *Autonomous operations*: As infrastructure becomes more autonomous, the operational framework controlling it will simultaneously mature. Tomorrow's autonomous operations will enable self-deploying infrastructure through automated procurement, deployment, configuration, optimization, and provisioning with minimal human intervention and oversight. Once live, operational services will automatically patch, upgrade, audit, license, and remediate issues. On the other end of the infrastructure lifecycle, autonomous operations will deprovision, sanitize, and retire aging infrastructure. "Over the next five years, we'll see the training wheels come off and IT operations will autonomously monitor, migrate, and remediate infrastructure issues," comments Rao.[12]

To summarize, the three primary evolutions in IT infrastructure require analogous changes to remove humans from the loop, eliminate boundaries between environments, and transition from reactive to proactive operations. In a word, tomorrow's IT operations will become invisible.

CHALLENGES

Transitioning to invisible IT operations will present many challenges, including

- *Trust*: The largest challenge will be trusting autonomous infrastructure and operations to procure, deploy, and operate an enterprise's key workflows reliably, economically, and responsibly. "Trust isn't binary," notes Rao. "The path forward will be an evolution from human-in-the-loop to human-alongside-the-loop to human-in-review before we take the training wheels off."[12]
- *Data gravity*: While invisible operations will transition workflows between environments, data gravity will prevent applications from moving outside the physical latency limits of their data storage. Caching and replication technologies will be developed to counter this, but gravitational data pull will have a profound effect on operational workflows, especially at the edge.
- *Security*: Invisible operations development will be underpinned by parallel advancements in security to protect enterprise data, infrastructure, users, and workflows from nefarious actors. This will be an arduous, expensive, and

complicated endeavor requiring new tools, techniques, and processes, especially at the edge.

The velocity and volume of evolving enterprise workloads are causing IT infrastructure to become more disaggregated, distributed, and predictive and, consequently, IT operations to become more ubiquitous, smart, and autonomous. These profound changes to the IT landscape will undoubtedly face challenges, but the drive for invisible operations will be irresistible.

REFERENCES

1. IEEE, "Heterogeneous integration roadmap," in *High Performance Computing and Data Centers*. Piscataway, NJ, USA: IEEE, 2021, Ch. 2, pp. 6–7.
2. S. Walleit, "Virtual storage: Definition, advantages, and a comprehensive guide." Parallels. Accessed: Nov. 28, 2022. [Online]. Available: https://www.parallels.com/blogs/ras/virtual-storage/
3. J. Niemöller, R. Szabó, A. Zahemszky, and D. Roeland, "Creating autonomous networks with intent-based closed loops," *Ericsson Technol. Rev.*, vol. 2022, no. 4, pp. 2–11, Apr. 2022, doi: 10.23919/ETR.2022.9904673.
4. S. Katukam, private communication, Jun. 2022.
5. J. Morris, "The pull of data gravity," *CIO*, Feb. 2022. [Online]. Available: https://www.cio.com/article/305285/the-pull-of-data-gravity.html#:~:text=Data%20gravity%20is%20the%20power,according%20to%20the%20IDC%20report
6. L. Shiff, "Popular IT service management (ITSM) frameworks," BMC, Houston, TX, USA, 2021. [Online]. Available: https://www.bmc.com/blogs/itsm-frameworks-popular/
7. D. Motiso. "A guide to ITSM frameworks: Definition, types and benefits." Indeed. Accessed: Nov. 24, 2022. [Online]. Available: https://www.indeed.com/career-advice/career-development/itsm-framework
8. R. Amarnath, "Why hybrid cloud data management makes good business sense," *Forbes*, Jun. 2022. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2022/06/02/why-hybrid-cloud-data-management-makes-good-business-sense/?sh=8886c73d78fe
9. S. Carey, "What is infrastructure as code? Automating your infrastructure builds," *Infoworld*, Dec. 2021. [Online]. Available: https://www.infoworld.com/article/3344382/what-is-infrastructure-as-code-automating-your-infrastructure-builds.html
10. S. Subhash, "Modernizing IT operations with AIOps," *Forbes*, Apr. 2022. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2022/04/20/modernizing-it-operations-with-aiops/?sh=79b9e5822338
11. R. Reddy, "How is predictive analytics transforming IT operations," *Acuvate*, Feb. 2021. [Online]. Available: https://acuvate.com/blog/predictive-analytics-it-operations/
12. G. Rao, private communication, Nov. 2022.
13. K. Wali, "Cloud computing vs fog computing vs edge computing: The future of IoT," *Analytics India Mag.*, Feb. 2022. [Online]. Available: https://analyticsindiamag.com/cloud-computing-vs-fog-computing-vs-edge-computing-the-future-of-iot/
14. E. Spears, "Everything as a grid – The data center infrastructure of tomorrow," *Data Center Post*, Sep. 2020. [Online]. Available: https://datacenterpost.com/everything-as-a-grid-the-data-center-infrastructure-of-tomorrow/#:~:text=In%20an%20%E2%80%9CEverything%20as%20a,a%20partner%20in%20energy%20transition

MARK CAMPBELL is the chief innovation officer at EVOTEK, San Diego, CA 92121 USA. Contact him at mark@evotek.com.

DEPARTMENT: GAMES

This article originally appeared in Computer vol. 56, no. 1, 2023

The Metaverse University

Michael Zyda, *University of Southern California*

The metaverse that is being created is starting from games but is not limited to games. There are scant educational programs to support this growing field. In this article, we propose the founding of The Metaverse University.

In 2005, I founded the Computer Science Games Program at the University of Southern California (USC) (the USC GamePipe Laboratory), and the purpose of that program was to provide a stream of engineers, artists, and gameplay designers to the rapidly growing games industry. Since 2005, the games industry has grown by a factor of five in size, and the program I created at USC has graduated some 4,000+ students into positions in the game and computing industries.

We are now in a rapidly changing environment where major efforts are being placed on creating the metaverse—technologies are quickly being created to support the metaverse and its application. The metaverse that is being created is starting from games but is not limited to games; there are applications—medical, engineering, and others—that are also demanding a place in the metaverse. There is not much university focus on the metaverse, and there are scant educational programs to support this growing field. Without an educational program focused on metaverse technologies and their application, this new field will not be able to grow.

THE METAVERSE UNIVERSITY

We propose the creation of a new educational entity called *The Metaverse University*, an educational university that will graduate engineers, artists, designers, and producers who can design and build our future metaverse technologies and applications. We plan on creating a university without departmental boundaries to eliminate the stove-piping issues we discovered in the stand-up of various games programs. Our model will be similar to that of Rockefeller University, where all faculty and students are just members of the university, not departments. Students will follow educational tracks—metaverse design, development, art, audio, and production. The last year for each student will be a year-long, team-based metaverse development project capstone course followed by a demo day for presentation of the developed technologies and applications.

WHAT DOES THE EDUCATIONAL PROGRAM LOOK LIKE?

The metaverse educational program has three major parts: metaverse development, metaverse design, and metaverse art and design (Figure 1). Each of these parts is briefly described in the following sections.

Metaverse development

Metaverse development is the engineering part of building the applications that allow users to move through and interact with the developed 3D graphics world. Until recently, building a 3D environment for a metaverse required strong, relatively low-level game engine programming. The programmer would decide to use the Epic Unreal or Unity engine and, typically, would find that the engine required significant software development before it could properly be used. Engines are typically not yet ready for much nonprogrammer development. The metaverse industry has created a number of platforms—platforms are typically built on top of a game or similar engine, and the

Digital Object Identifier 10.1109/MC.2022.3218389
Date of current version: 9 January 2023

purpose of having a platform is to reduce the complexity of metaverse application development. Some of the platforms available include Nvidia's Omniverse, Unity's Metaverse Toolkit, Intel's Metaverse Toolkit, Microsoft's Metaverse Toolkit, Epic's Fortnite Creator, and Roblox. To be able to graduate strong metaverse developers, we put the following courses into the metaverse development degree program:

- Metaverse Design Workshop
- 3D Computer Graphics and Rendering
- Metaverse Platforms and Tools
- Metaverse Modeling and Simulation
- Metaverse Networking Infrastructures
- Machine Learning for the Metaverse
- Building the Human-Intelligent Metaverse.

The Metaverse Design Workshop class is taken by all students in all degree tracks at the Metaverse University. The purpose of the class is to teach our students how to think in 3D for metaverse development and how to generate an appropriate metaverse design document (MDD) before cutting code. This class has the students develop a runnable prototype using one of the available industry platforms once their MDD has been completed and approved. The MDD is not unlike the game design document used in the games industry except that designing an application that will be in 3D—with a 3D interface and all of its digital art in 3D—is somewhat more difficult than the 2D side-scroller usually found in games programs.

In the metaverse development program, we have to make a decision as to what appropriate foundational courses should be required. One such course is a solid course: 3D Computer Graphics and Rendering. Most university computer science departments have this course in some fashion. Universities that are hopelessly out of date will have the students in this course turn lines and triangles into pixels in an in-memory frame buffer. This is out of date, as this is all done in graphics hardware today—it might have been appropriate, say, in a 1979 version of this course. Today, for the metaverse development realm, there is probably high-level viewing and matrix transformation functionality, so students who will be building at that level should understand such mathematics at the symbolic level as presented in text, such as that by Hughes et al.[1] Additionally, students who come out of this course should thoroughly understand shaders and rendering pipelines as are provided by today's graphics hardware.

The Metaverse Platforms and Tools course (Figure 2) starts out by covering the Open Metaverse standard toolsets and definitional files that should be utilized in constructing your metaverse application. The next part of the course is to learn about game engines—say, the Epic Unreal and Unity engines—to understand how those engines reach out efficiently to the graphics pipeline for the proposed hardware. This course then moves on to the various available industry metaverse platforms that may be of interest to the

COMMENTS?

If you have comments about this article, or topics or references I should have cited or you want to rant back to me on why what I say is nonsense, I want to hear. Every time we finish one of these columns, and it goes to print, what I'm going to do is get it up online and maybe point to it at my Facebook (mikezyda) and my LinkedIn (mikezyda) pages so that I can receive comments from you. Maybe we'll react to some of those comments in future columns or online to enlighten you in real time! This is the "Games" column. You have a wonderful day.

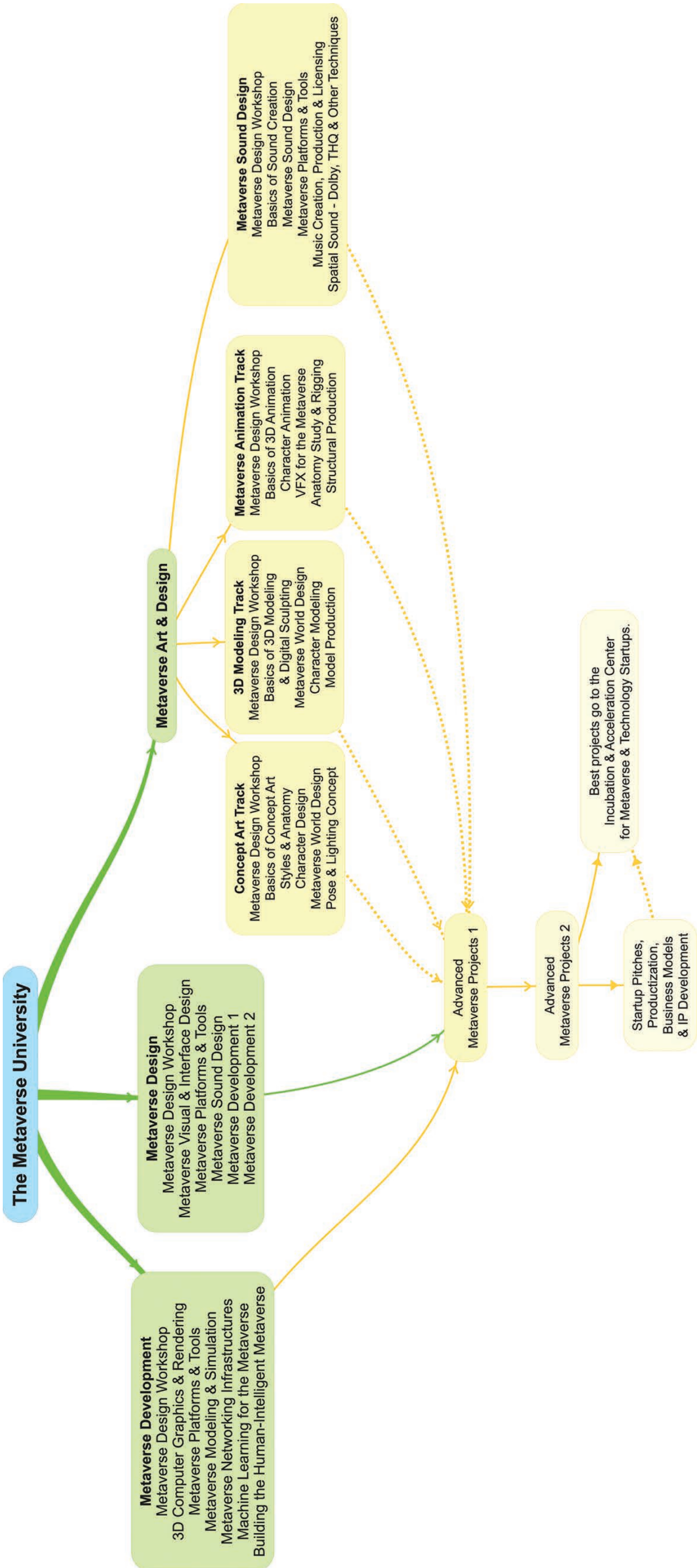

FIGURE 1. The Metaverse University.

student. Nvidia's Omniverse is an outstanding choice for the student interested in creating a metaverse that requires high-performance GPUs, as are manufactured by Nvidia. If the student is interested in a platform that works on pretty much any hardware, then the Microsoft/Intel/Unity metaverse toolkits are the way to go. At an even higher level of functionality that is less complex programmatically, the Epic Fortnite Creator and Roblox platforms are what to utilize. Therefore, the purpose of this class for the student is to understand how to utilize a modern metaverse platform as well as how to use a game engine that may be under that platform—the engineer may have to add code at the engine level for a special feature not in the higher level platform.

The Metaverse Modeling and Simulation course (Figure 3) begins with time management in real-time simulation—a complete integrated model of how time operates in the individual metaverse is essential. A classic article on this is Time, Clocks, and the Ordering of Events in a Distributed System.[2] We cover discrete event simulation. We then move on to physics engines and real-time collision detection. The end, nonsmall topic for this course is digital twins—real-time virtual representations of real-world physical systems or processes.

The Metaverse Networking Infrastructures course (Figure 4) is fundamental in that the expectation for all metaverses is that there will be multiple people in each metaverse as well as standards and methods for moving your virtual character from one

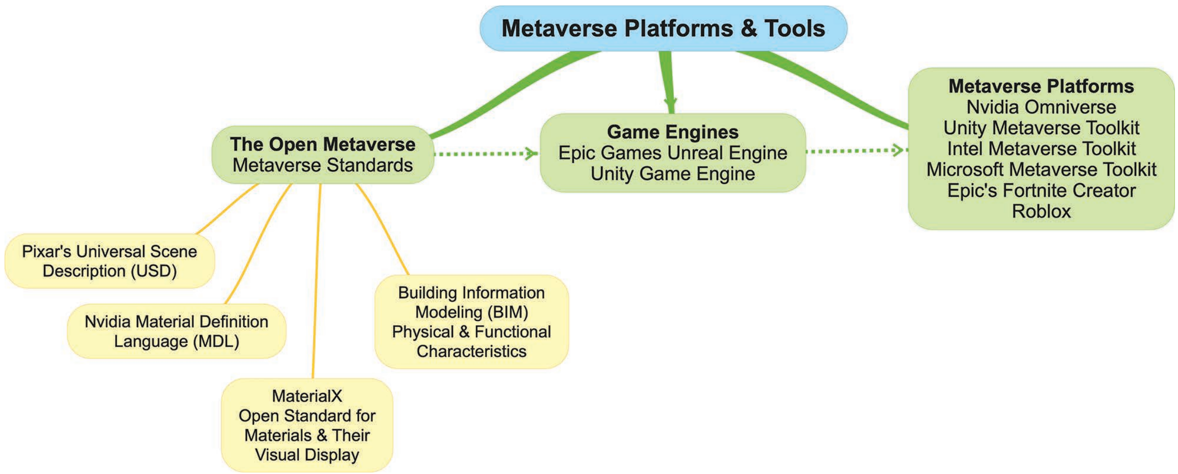

FIGURE 2. The Metaverse Platforms and Tools course.

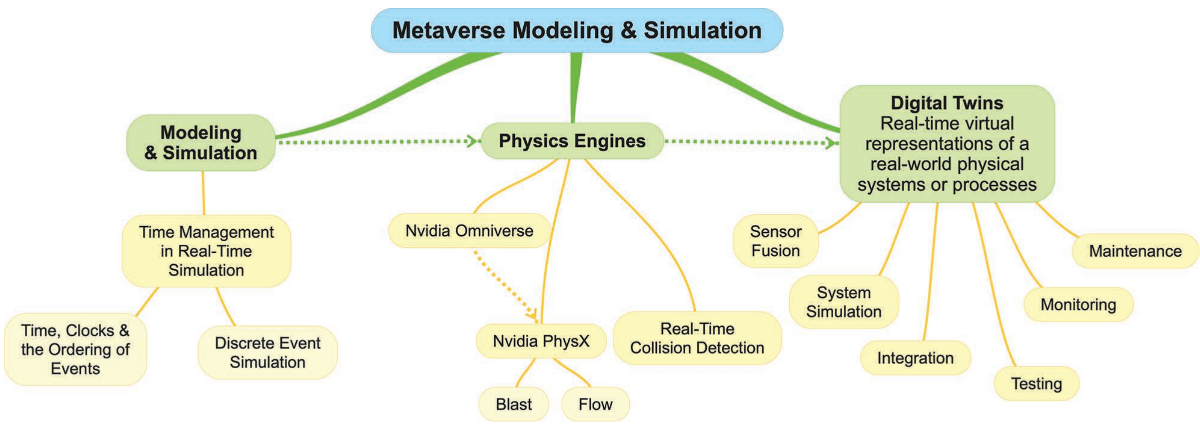

FIGURE 3. The Metaverse Modeling and Simulation course.

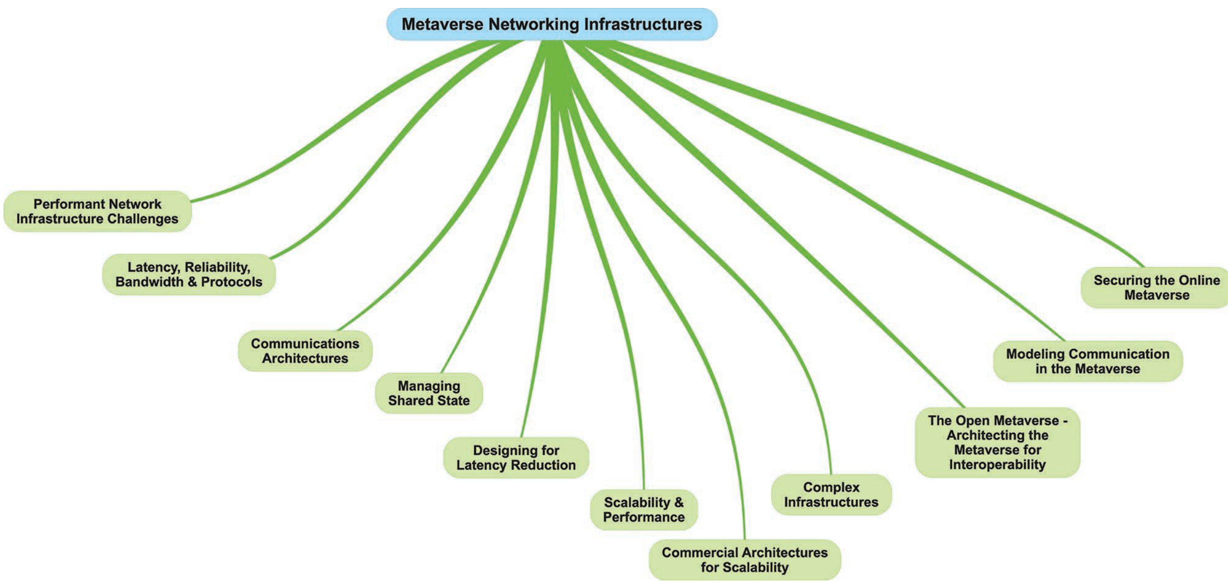

FIGURE 4. The Metaverse Networking Infrastructures course

metaverse to another. Since those standards and interoperability do not yet exist, this course covers everything the engineer requires understanding in, including latency, reliability, bandwidth, protocols, communications architectures, managing shared states, designing for latency reduction, scalability and performance, complex infrastructures, Open Metaverse architectures, communication modeling, and securing the online metaverse. The information from this course is critical to the secure and performant operation of every metaverse and is not just something, at the moment, that "comes with the game engine." These issues were well discussed in a previous edition of this column, and the interested reader is pointed toward that article.[3]

The Machine Learning for the Metaverse course is conducted at the application level as opposed to the mathematical fundamentals of machine learning (ML). The application of ML to metaverse development is an important focus for the metaverse industry. This course covers the use of ML for artificial intelligence bot creation, metaverse interaction analysis, and real-time metaverse user understanding. As metaverse platforms move beyond their game engine origins, embedded ML for the operation, analysis, and understanding of metaverse interaction will become a prominent and important component.

The Building the Human-Intelligent Metaverse course provides an understanding of the available biosignals generated by humans, sensors that can read those biosignals, and human modalities that can be computed from biosignals. Topics covered include human-generated biosignals; emotion states; emotion sensors and computational models for interpretation; emotional state vectors; detected emotions and probabilities; sensor bandwidth, power, and computation requirements; physical state sensors and their computational models; mental state sensors and their computational models; sensor fusion; ML for computing human states and modalities from sensed biosignals; and computed human modalities.[4]

METAVERSE DESIGN

Metaverse design is the degree program that teaches creatives how to draft an MDD, a document that is written before code is developed describing how the metaverse app goes together, how its user interface works, and any other important design features. The courses taken by students in the metaverse design program are the following:

- Metaverse Design Workshop
- Metaverse Visual and Interface Design
- Metaverse Platforms and Tools
- Metaverse Sound Design
- Metaverse Development 1
- Metaverse Development 2.

We already defined the Metaverse Design Workshop class earlier in the "Metaverse Development" section and will not repeat that material here. Remember, everyone in each track at The Metaverse University takes the Metaverse Design Workshop course.

The Metaverse Visual and Interface Design course is one of the most important. In this course, design students are taught how to think conceptually in 3D such that a coherent, unified 3D interface for the metaverse app can be specified. This course also teaches how to think about software architecturally such that the overall 3D interface defined is used in the same way for all parts of the specified app and, in fact, for all apps of the series expected to be developed.

The Metaverse Platforms and Tools course we discussed earlier in the Metaverse Development section.

The Metaverse Sound Design course is a tool- and software-level course that teaches the designer how to create sounds and utilize them in a metaverse platform such that we "see a sound and hear a sound," as sound designers are wont to say. This course additionally covers music integration into the metaverse platform. Again, sound integration into a 3D metaverse means that it is the metaverse user's expectation that sounds will be appropriate to their spatial location and to the characteristics of the 3D virtual materials in that part of the metaverse.

The Metaverse Development 1 and 2 courses are courses where the students form small teams for the creation, over the course of a semester, of a prototype metaverse implementation selected by each student team. The idea is to get the students used to building rapid prototypes of metaverse apps with high-level platforms. These two courses must be completed before the students move into the Advanced Metaverse Projects course.

Metaverse art and design

The metaverse art and design degree program focuses on the creation of art and sound assets for use in the development of metaverse apps. There are four tracks in the metaverse art and design program: concept art, 3D modeling, metaverse animation and visual effects (VFX), and metaverse sound design.

Concept art track. The concept art track focuses on teaching the student how to make concept art that properly communicates to the developers of the metaverse application what that particular metaverse should, in the end, look like. It is always nice to know expectations before software and 3D assets are cut.

The concept art track student, then, takes the following courses:

- Metaverse Design Workshop
- Basics of Concept Art
- Styles and Anatomy
- Character Design
- Metaverse World Design
- Pose and Lighting Concept.

Basics of Concept Art is exactly that. This course teaches the student how to create concept art that visually displays the vision as expressed in writing by the metaverse designer. The Styles and Anatomy course teaches the student how to create a coherent style for the concept art as well as the appropriate anatomy of human and other figures to be shown in the concept art. Character Design is the course for building the concept art of prominent characters in the metaverse. Metaverse World Design is the course that teaches how to create the 3D worlds where your metaverse app takes place. A concept art student in that course will focus on concepts for such 3D worlds. Pose and Lighting Concept is the course that teaches the artist how to pose and light characters and the world model.

3D modeling track. The 3D modeling track focuses on teaching students how to build all of the 3D parts of the metaverse app to be developed. The courses in this track are the following:

- Metaverse Design Workshop
- Basics of 3D Modeling and Digital Sculpting
- Metaverse World Design
- Character Modeling
- Model Production.

The Basics of 3D Modeling and Digital Sculpting course is exactly that, using appropriate digital tools. The Metaverse World Design course we detailed earlier. Character Modeling is how the 3D characters are designed and built for our metaverse. Model Production is a course on how to create a pipeline process for 3D art production for a metaverse development project.

Metaverse animation and VFX track. The metaverse animation and VFX track is how we give life to the characters and the world we have developed for our metaverse. The courses in that track are the following:

- Metaverse Design Workshop
- Basics of 3D Animation
- Character Animation
- VFX for the Metaverse
- Anatomy Study and Rigging
- Structural Production.

Basics of 3D Animation teaches the student classical animation methods for getting the characters and the world in motion. Character Animation is how we bring our characters to life and provide their visage with emotion. VFX for the Metaverse is how we use special effects to animate the world and make it have a particular ambient emotion. Anatomy Study and Rigging is how we create a framework inside of our 3D characters that we can manipulate via our available user interface. Structural Production is how we create a pipeline of 3D modeling, animation, and VFX for all of the characters/worlds and all of the artists working on our production.

Metaverse sound design track. The metaverse sound design track is how we create artists who can develop and integrate sound into our metaverse productions. The courses in that track are the following:

- Metaverse Design Workshop
- Basics of Sound Creation
- Metaverse Sound Design
- Metaverse Platforms and Tools
- Music Creation, Production and Licensing
- Spatial Sound: Dolby, THQ, and Other Techniques

The course titles are accurate as to our expectation as to what will be covered. Building sound and music for use in the metaverse is critical for creating the illusion of life in that world.

ADVANCED METAVERSE PROJECTS

All students in our program take the Advanced Metaverse 1 and 2 courses. These courses are the team-based production of a significant metaverse application or technology over the course of two semesters. The spring semester before this year-long project is where the students pitch their proposed plan as to what they would like to build over the course of a year. University faculty and staff watch those pitches and develop a greenlight list as to what is approved for production.

BUILDING SOUND AND MUSIC FOR USE IN THE METAVERSE IS CRITICAL FOR CREATING THE ILLUSION OF LIFE IN THAT WORLD.

Start-up pitches, productization, business models, and intellectual property development

One of the final courses all of our students must take covers how to create a successful start-up pitch, how to productize developed prototypes, business models for marketing your prototype, and intellectual property development and protection. We plan on teaching our students to fish.

Incubation and Acceleration Center

The best projects from our students in the Advanced Metaverse Projects course have the potential to be accepted into the Incubation and Acceleration Center. In that center, we provide space and resources for student teams to advance or complete their metaverse application. We additionally provide them guidance toward potential investment partners.

WEB3 AND NFTs ...

Whenever I mention to people that I am planning the creation of a new university, The Metaverse University, either I get great understanding from real-tech people who have been in the 3D virtual environment space for some time, or I get the blockchain/Web3/NFT fanboys/fangirls telling me that the metaverse is just about Web3, describing how they will soon become rich beyond their wildest dreams, and asking me what I think about crypto. I usually delete any person who chats with me about crypto online, but I do believe we should perhaps study the whole blockchain/Web3/NFT space to see if there is some nugget that is not fully and foolishly useless.

Some comments from game industry greats are aligned with me on this—the following two are from Gabe Newell, CEO of Valve:

- *Gabe Newell on blockchain and NFTs*:

  "The things that were being done were super sketchy. And there was some illegal shit that was going on behind the scenes, and you're just like, yeah, this is bad. Blockchains as a technology are a great technology, that the ways in which has been utilized are currently are all pretty sketchy. And you sort of want to stay away from that. … The people who are currently active in that space are not usually good actors."[5]

- *Gabe Newell on the metaverse*:

  "There's a bunch of get rich quick schemes around metaverse. Most of the people who are talking about metaverse have absolutely no idea what they're talking about. And they've apparently never played an MMO. They're like, "Oh, you'll have this customizable avatar." And it's like, well … go into La Noscea in *Final Fantasy 14* and tell me that this isn't a solved problem

> from a decade ago, not some fabulous thing that you're, you know, inventing."[5]

Another article, "The Boom and the Bust: How NFTs Went the Way of Beanie Babies," is also worth reading:

> "In early 2021, NFTs were pitched as a way for artists to make life-changing money. All you needed was a token! So artists raced out to buy ether so they could mint NFTs of their work. Only, the money didn't pour in—unless your name was Beeple."
>
> "Now all that is left of the once overly hyped NFT market is dust in the wind. NFTs are becoming an unattractive piece of history. People aren't trading them, and the only ones talking about them are those trying to sell them."[6]

On one of my presentations to Nvidia on founding this new university, I had a slide on our proposed adjacent research university, and Nvidia's comment was that they all thought blockchain/Web3/NFTs was all just a big scam and that I should take that off my research slide. I think I am going to leave it on my research slide so that it reminds me that perhaps I ought to take a closer look—just in case there is a winning lottery ticket inside.[7]

All of this is a proposal to build The Metaverse University. We have developed the course plan for the educational program and have had initial discussions with potential faculty. We have developed an operating budget for the first three years. We are looking at various sites for the stand-up of The Metaverse University (Los Angeles, Seoul, and Dubai) and are looking at acquiring the first space, a space that can educate students in Los Angeles.

We are looking for a key sponsor partner for the first three years of operation for The Metaverse University. We believe that, by the start of the fourth year of operation, tuition and fees will be able to cover this university's operating costs.

ACKNOWLEDGMENT

There are many people who have been greatly supportive of this plan to create The Metaverse University: Mike Setlich, Thomas Lee, St John Colon, Scott Easley, Vangelis Lympouridis, Qingyun Ma, S.C. Mero, Rudy Darken, Don Brutzman, and many, many former students and colleagues with whom I have had great interactions. "To boldly go where no one has gone before" (—*Star Trek*, of course).

REFERENCES

1. J. Hughes J. D. Foley, A. van Dam, and S. K. Feiner, *Computer Graphics: Principles and Practice*. Reading, MA, USA: Addison-Wesley, 2013.
2. L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Commun. ACM*, vol. 21, no. 7, pp. 558–565, Jul. 1978, doi: 10.1145/359545.359563.
3. M. Zyda, "Let's rename everything 'the metaverse!'," *Computer*, vol. 55, no. 3, pp. 124–129, Mar. 2022, doi: 10.1109/MC.2021.3130480.
4. M. Zyda, "Building a human-intelligent metaverse," *Computer*, vol. 55, no. 9, pp. 120–128, Sep. 2022, doi: 10.1109/MC.2022.3182035.
5. P. Tassi, "Valve's Gabe Newell takes a flamethrower to the metaverse and NFTs," *Forbes*, Feb. 2022. Accessed: Sep. 24, 2022. [Online]. Available: https://www.forbes.com/sites/paultassi/2022/02/26/valves-gabe-newell-takes-a-flamethrower-to-the-metaverse-and-nfts/?sh=103ab6cd1a6a&fbclid=IwAR162VhWiIC4meWdRdx0RbfTDu_VE2ZdLGBX8mnPgFeklBMzFQ9WEI0CFGM
6. A. Castor. "The boom and the bust: How NFTs went the way of beanie babies." Artnet. Accessed: Sep. 24, 2022. [Online]. Available: https://news.artnet.com/market/nfts-boom-bust-and-backlash-2176451
7. S. Alpher and A. Ashraf. "Compute north files for bankruptcy as crypto-mining data center owes up to $500M." CoinDesk. Accessed: Sep. 24, 2022. [Online]. Available: https://www.coindesk.com/business/2022/09/22/crypto-mining-data-center-provider-compute-north-files-for-bankruptcy-protection/?fbclid=IwAR1dAcnEql0iwr1KJl4x3Gz1Db0D_9XQJ84tOMHDVYtxTsRMwqZswKo-1tk

MICHAEL ZYDA is the founding director of the Computer Science Games Program and a professor of engineering practice in the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA. Contact him at zyda@mikezyda.com.

EDITOR: Michael Zyda, University of Southern California, zyda@usc.edu

DEPARTMENT: GAMES

This article originally appeared in Computer vol. 55, no. 5, 2022

How Do I Get a Position in the Games Industry? The FAQ

Michael Zyda, *University of Southern California*

The question most often asked of me at all levels is, "How do I get a position in the games industry?" I will attempt to answer this in the kindest way possible using my sage years of experience.

After 17 years at the University of Southern California (USC) of founding, running, and building the games program, the question most often asked of me is, "How do I get a position in the games industry?" This is being asked at all levels—by seniors in high school, soon-to-be-college-graduates who have not taken any game-building classes, and many others who started down other career paths that ended before they did. I attempt to answer it in the kindest way possible using my sage years of experience. There are many subquestions that are part of this, and I list and answer them as smartly as I know how.

First of all, you really have to want to get a position in the games industry and be compelled to do this—if that is you, then great! Read on! If you are just thinking about it but studying/working in something else, it's probably not for you. In modern times, getting a position in the games industry means some years of study at the right university and experience with building games in teams. It used to be that all you had to do was take beginning programming and data structures courses and then drop out of school to found your own studio. That is rarely true now and not a recommended pathway to the games industry, says the professor who teaches games courses in a university… . So, let's start out by assuming you really want to do this, and you are ready for some education!

Digital Object Identifier 10.1109/MC.2022.3151459
Date of current version: 6 May 2022

WHAT GAMES PROGRAM SHOULD I ATTEND?

This question gets asked of me all the time, even though I have been at USC for 17 years and founded the games program there. I usually do not answer this with, "USC, of course"; I usually try to provide guidance as to what the potential student should look for in a games program.

The way I usually start out is to let the questioner know where hiring is happening in the games industry. First of all, one of the constants in the games industry is that the demand for engineers/programmers is always strong:

- 60% of the demand in hiring for positions in the games industry is for engineers who know how to build games and work in cross-disciplinary teams
- 30% of the demand is for artists who know how to utilize the tools used by the games industry and produce art for the cross-disciplinary team
- 5% of the demand is for creatives in music and sound production for games
- 5% of the demand is for gameplay designers—and, for the most part, that demand is for seasoned designers, not fresh university graduates.

Cross-disciplinary teams means programmers, artists, gameplay designers, musicians, and sound engineers. Notice that I am not including business, legal, and marketing people, who are also essential. I am just going to focus on the actual game developers,

not the corporate-side leeches. No. I really appreciate everyone!

SOME GAME SCHOOL PROGRAM QUALIFICATIONS

Engineers/Programmers

When I say engineers/programmers, the school you are looking for ought to be teaching you C++ as a first language, followed by a data structures in C++ course, followed by a team-based project of some sort where your C++ skills are utilized. If the school you are considering is still teaching these courses in Java, you ought to run away. If the hiring people in the games industry hear that you learned Java as a programming language before you learned C++, they will terminate the interview and not consider you further. That's just the way it is. Also, C# is not C++, and, while skills in C# are great for building side-scrollers for the mobile market, C# will not help you get a position in a AAA-title-building game development studio. I pointer you to C++! programmer humor.

That program also needs to include a solid course in networking, preferably at the applications layer, and a solid course on operating systems. All AAA titles are networked and need responsive software built with an understanding of operating systems. These two courses are essential, and any program on game development that does not have this solid base for the engineers ought to be avoided, as the games industry hiring people will expect that you have had these. They will test you—you cannot just skip these courses and get found out later in the interview.

Another course engineers typically take is game engine development—this is not a course on how to use a game engine but how to build one. This course has been stuck inside of most games programs since their start circa 2005—when the expectation was that students would build their own game engine because there were not any good open source ones then. Now, a real game engine is a big piece of software. For example, Unreal Engine has more than 16 million lines of code, and your custom code on top of that engine will be 200,000–1,000,000 additional lines by the time your game ships.[1] Epic has probably spent more than US$1 billion developing this engine.

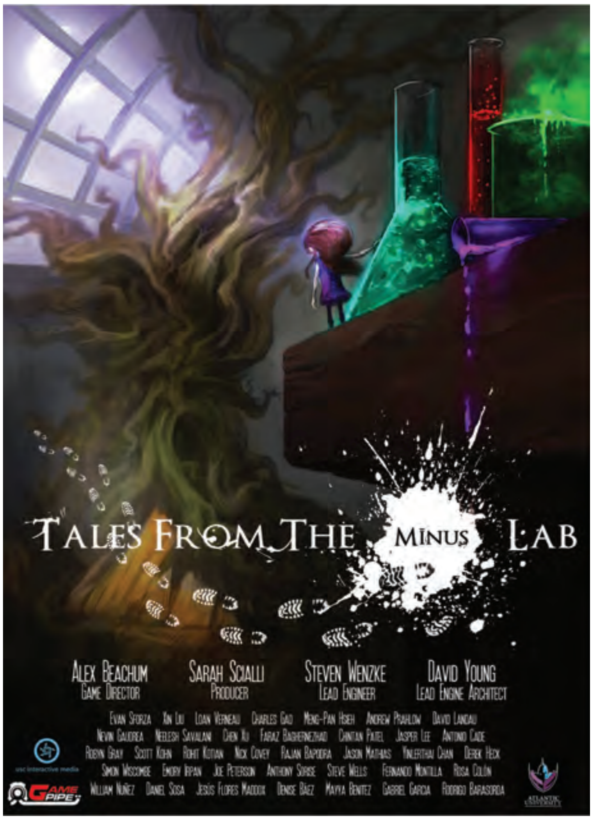

FIGURE 1. The *Tales From the Minus Lab* poster from the USC Advanced Games Project course.

Therefore, the expectation that you will build a fully functioning game engine inside of your university course is a bit off. The closest I have ever seen a student team get to developing its own game engine was during our year-long advanced game projects course, where one half of the development team built its game, *Tales From the Minus Lab* (Figure 1), on top of the Unity game engine, while the other half developed a game engine that supported the functions Unity provided to the game. This could have been a train wreck, but the team was awesome and delivered both a game and a game engine. Most game engine development courses just task you with building the graphics rendering pipeline part of the game engine in C++. That is probably an excellent experience, and I highly suggest that you find a program that requires it.

Once you are so equipped with appropriate programming/engineering skills, you should then take a course on game design, which ought to use your programming skills to actually build something, not just hand you scissors, cardboard, and markers for you to paper-prototype something over a whole semester. Don't waste your time—build it!

The next thing you ought to do in this program is spend a semester working in a team, designing and prototyping a game that you can pitch for development over a two-semester period—we call this the advanced game projects (AGP) course. This ought to be a simulation of an industrial team, meaning you build a game over an academic year that is so unique that, when you show it to hiring people, they will immediately want to know your role in its development and, maybe, even consider hiring your entire team. That is the goal—it might be a reach, but that is what you should be shooting for.

Now, today, it is very easy to download game starter kits from Unity or Epic, but, if all you are going to do is build yet another side-scroller, no one is going to care. You need to build something that shows off your technological prowess and game design skills. If your game doesn't have networking, 3D characters, cool shaders, and something that grabs a players' need for gameplay, then you will not get far with the hiring people for the AAA-title games industry.

Artists

The most important lesson is that there is not just one kind of artist in the games industry—remember that 30% of the hiring demand is for artists. Roughly, there are concept artists, 3D modelers (character modelers and environment modelers), texture artists, lighting artists, and animators. Therefore, if you are selecting a games program, you ought to choose one that has a substantial offering in game art and design that joins with the engineers and gameplay designers in a year-long AGP course.

Concept artists are required at the start of most game development to sketch out or model what the game will maybe look like. This is nice to have at the beginning, as it has a large impact on the selection of the technology to be used in building the game. Concept artists are specialized—they have skills that sell the project before production, when the money people have not yet said yes. They are also great for putting together a storyboard for the illustration of a complex part of the game during development.

There are two kinds of 3D modelers: character and environment modelers. Character modelers build the 3D person/animal/robot you want for your game. Animators rig those 3D characters so they can move. Sometimes, your character modeler can rig, but model rigging is highly specialized, and it may be another person with different talents.

Environment modelers build the 3D models of the gamescape where the game takes place. They build buildings/mountains/forests where our 3D characters can frolic/run/shootor whatever they do in such scapes. The tool used by environment modelers is the same one character modelers and animators use, Autodesk's Maya.

Texture artists take the 3D models and apply a texture to the polygons that make up the 3D models. Texture artists tend to act strangely if you give them a digital camera to use—they will take pictures of paint peeling, dirt, sand, mud on your car, and just about anything that they might be tasked to apply onto the 3D models. I appreciate them greatly, as they make the 3D models/worlds look dirty and used, like the real world. Texture artists often work with lighting artists to add shaders to the polygons of the model so those textures are applied. Lighting artists light the scene, the entire 3D world and all its parts, with the hope that that lighting runs in real time, with the 3D modelers getting the blame if they use too many polygons for that world.

It is a rare games school that has such a comprehensive game art and design program—usually, game schools have to reach out to art schools and beg them to please do art for the games being built in the AGP course. Such relationships are valuable and delicate, and they should be treated with the utmost care.

Check if the games school you are considering has 1) a game art and design program or 2) strong relationships with outside art schools. If neither of these is true, then consider looking elsewhere. The alternative is that all your games use "Asset Store downloaded" models, which look nice individually but never look well together, as they rarely match the concept art you have, in either digital form or your head. Access to real artists makes the games school great—be there.

Gameplay Designers

Gameplay design is how you create the story and interactions the player of the game performs. There is the overall story, sometimes called the *backstory*, and then the story for the player, meaning the player's role in that story and how he or she executes that role through some kind of interaction—button presses, keystrokes, and so on.

Game schools that teach you how to do gameplay design are going to teach you the history and vocabulary of gameplay, the various genres of games that have been created, and how you produce a game design document (GDD) that becomes the guide for your development team, a guide that needs to be continuously updated as game production proceeds. GDDs have sections for the introduction, audience, platform, system requirements, key features, reference games, gameplay overview, gameplay elements, gameplay verbs, gameplay philosophy, pillars of gameplay, what-this-game-is-not disclaimer, gameplay overview flowchart, win/lose conditions, replay value considerations, user interface (UI), gameplay scope chart, player character discussion, game world structure, levels, story overview, full narrative, and one-page production plan.

The introduction is a brief paragraph on what the game is about—the succinct game name and what do you do in that game. The audience is whom the game for and their ages—you want to build your game for a particular audience, and you need to know that before you start building it or drafting the rest of the GDD. The platform is what kind of device the game being built for, and system requirements are any special requirements for that platform—if you can't answer this in the GDD, stop.

The key features are three or four bullet points on what is interesting about the game you are proposing—if you cannot think of these, then why are you suggesting building this game? Do we really need yet another 2D side-scroller? I believe we do not and, in fact, suggest that Unity and Epic remove that starter kit from their websites! Please.

Reference games are the games that have influenced you to draft your GDD—they might be reference games in that "your game will play like them," "your art style will be similar to them," or "your story will be reminiscent of them." Hopefully, it will not be "the same game as xxx with different character skins," a 2D side-scroller, or a tower defense—I don't like tower defenses, either.

The gameplay overview has two parts—one is a single-sentence log line that tells you succinctly what the game is about. If you need two sentences, delete the second one. The other thing the gameplay overview section has is about three or four paragraphs that provide a high-level overview of your game's story and the character's role in it. I know that writing the short story is much harder than writing a novel, but you must resist making this overview overly overt.

The gameplay elements section is a short bullet point list of things like exploration, puzzle-solving, and swordplay. These are high-level things your player will be doing. Gameplay verbs are the things the player can do to experience the gameplay elements with the press of a button on the UI—walk, examine, pick up, use, and swing axe.

The gameplay philosophy is the overall philosophy of the development team with respect to the proposed game: "Our game seeks to innovate in the first-person shooter realm by drawing indelible chalk marks around each passed virtual character until the original sidewalk is no longer visible," or something like that. Usually, an additional five to six paragraphs are provided to explain this innovation to convince the green-light committee or investors that this is not just another side-scroller.

The pillars of gameplay section is a short bulleted list of how we are going to deliver on our promise of innovation in the game. "What this game is not" is a disclaimer to the reader (the development team, investors, and so on) to assure them that we are not creating yet another 2D side-scroller; we are building a significant 3D slice of the global metaverse that will

COMMENTS?

If you have comments about this article, or topics or references I should have cited or you want to rant back to me on why what I say is nonsense, I want to hear. Every time we finish one of these columns, and it goes to print, what I'm going to do is get it up online and maybe point to it at my Facebook (mikezyda) and my LinkedIn (mikezyda) pages so that I can receive comments from you. Maybe we'll react to some of those comments in future columns or online to enlighten you in real time! This is the "Games" column. You have a wonderful day!

surely make us all entertained and wealthy beyond our wildest dreams, along with a signature in blood.

The gameplay overview flowchart is a block diagram that shows how the various levels of the game are connected. Levels are complete, stand-alone parts of the game that are entered via success at another level or from the start of the game. Each level, typically, has a win/lose condition that lets you go on to either the next level/game end or the respawn location when you have lost the level. Each game must have an overall win/lose report when you get to the end of all levels or the end of all experience when it has been determined that you have completely lost.

The replay considerations section is the part of the GDD where you indicate either that you can only play the game once through, as things are exposed to you by the end of the game, or that you can play again and receive a new experience. The UI is where the buttons you press are and how they activate the gameplay verbs. In the UI are also informational displays that tell you your score or health so that you know if you are about to experience a miserable and humiliating virtual pause in your online entertainment.

The gameplay scope chart has way more detail than the gameplay overview chart. Its purpose is to provide an understanding of the complexity of what needs to be built. The player character discussion tells the development team the point of view of the character—first or third person—and how the player relates to the character in the overall narrative of the game.

The game world structure is a detailed narrative on what the hopefully 3D world of the game entails and how the characters move through it. Each level is broken out as to what it looks like and how it integrates into the overall game world.

The story provides the context for the game and a detailed understanding of how the player character moves through and operates inside of that story. There is an overview as well as a full narrative, with as many details as are useful for the development of the game.

There is a one-page production plan that indicates how many expected days each major task should take with respect to each specialized person involved in the game's development. The game producer who puts this together and does a good job has a position for life. If the game ends up taking eight years to produce after all of the development money has run out, then that producer might consider a different career in life.

That is what a GDD contains, and your mileage may vary. Gameplay designers are taught how to draft these through multiple projects in their educational program, sometimes testing their ideas with paper prototypes and, more frequently nowadays, via quick prototypes using starter kits downloaded from the Unity or Epic websites. The game school that you choose should provide you with some notion of how they provide you this education and as to how many of these gameplay designs have gone on to be deployed inside of hit commercial games.

Gameplay designers who do well in their educational programs rarely move directly into an industry role as lead designer. They typically enter industry as a level designer; technical designer; or, sometimes, assistant producer. Level and technical designer roles require skills beyond just gameplay design, and it is recommended that that additional education be pursued.

THE AGP COURSE

Now, at all of the schools you are looking at, one thing you want to make sure is that there actually is a 12- to 14-month long projects course, maybe with the title "AGP." The structure of this course is that it is a two-semester-long development project—say, the fall and spring semesters—with the game designs selected for production in the previous academic year's spring semester. At USC, we typically receive about 20 GDD submissions, and a faculty and industry committee

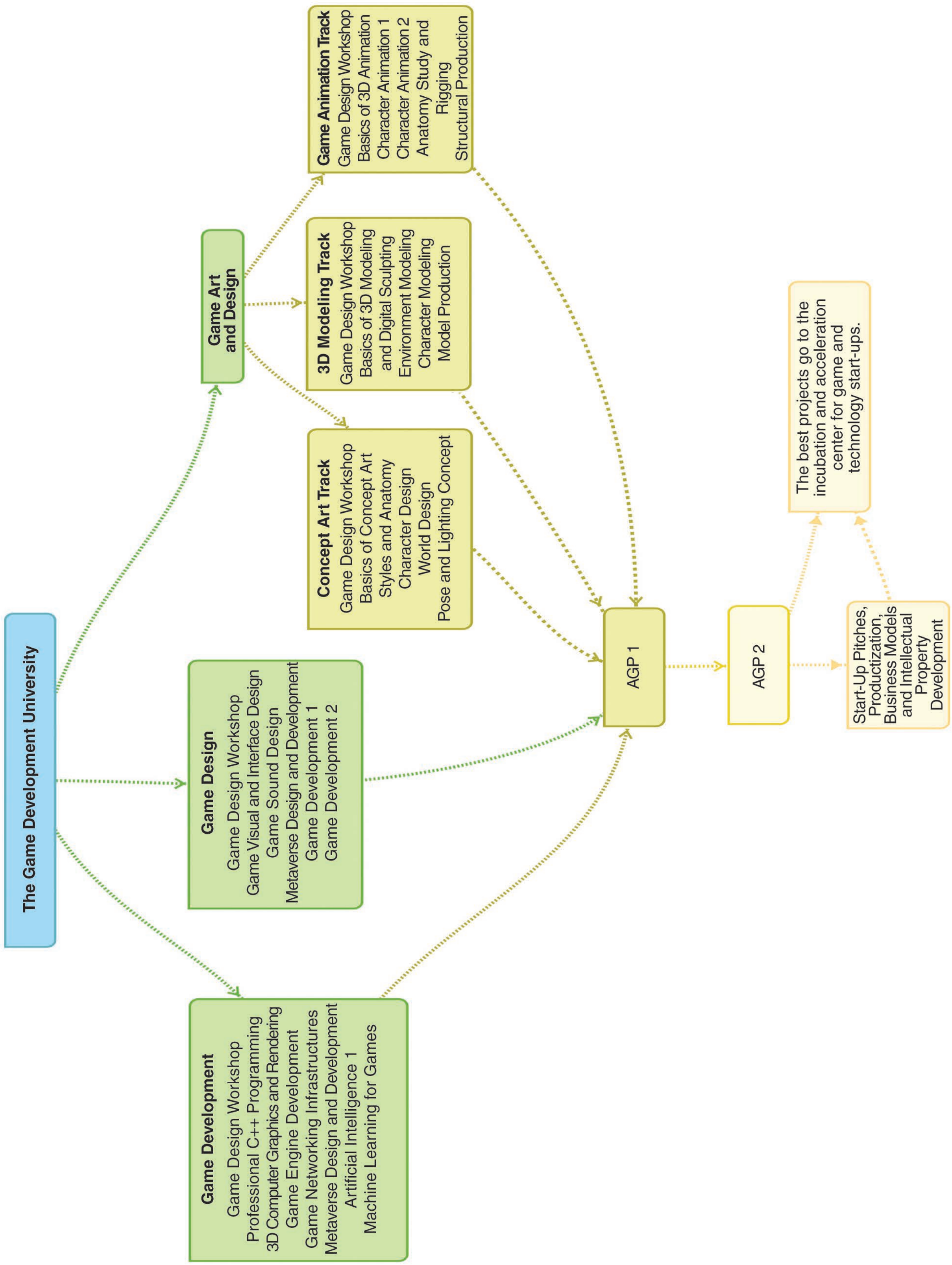

FIGURE 2. A prototype educational program for the Game Development University.

gets this down to six or seven finalists for production. Winning teams in the AGP course receive lab space; a team of student developers—maybe 20–65 students; and oversight from faculty with experiences in gameplay design, engineering, art, music, and sound. This course has run from 2005 to the present and has placed some 3,000–4,000 students into positions in the games industry. Look for a school like that and have that kind of experience, and it should be rather straightforward for you to get a position in the games industry.[2]

DEMO DAY, EXPO, WHATEVER THE NAME—SHOWING OFF YOUR GAMES TO INDUSTRY

Whatever games school program you are considering ought to have something called *Demo Day* or *Games Expo*, a biannual event where student-built games are shown off to the games industry. I say *biannual* meaning that there is a demo day at the end of the fall semester and one at the end of the spring semester. The fall demo day is the best one for hiring and internships—most such decisions are made during the time period from December through March. Therefore, showing off great projects at a demo day in December is essential. The spring demo day is also nice, as it is a great way to celebrate the game school's experience for its graduates—it is important to note that, by the end of the spring semester, all of the internship and hiring positions are already filled, and little in the way of hiring will occur at that end-of-spring-semester event.

Now that you know these demo day experiences exist, attend them for the game schools you are considering. They will get you excited about the game development career path and let you know about limitations in the game school's educational program. If most of the games are side-scrollers, maybe find another school that is pushing the boundaries of technology and game development. Figure 2 is provided as a reference for what you should be looking for.

ADDITIONAL EVALUATIONS FOR GAME SCHOOL SELECTION

Once you have studied the educational programs at the game schools you are interested in, you next need to consider the people who run them—you want to learn from those who have designed, developed, and shipped hit games. You don't want to learn game building from faculty who have not had commercial game-building experience, nor do you want to learn from someone whose primary experience in the games industry was to bankrupt the studio and put 800 developers out of work. The best way to discover something about the faculty is look them all up on LinkedIn and see who is best connected to the hiring people and leads in the games industry. The last time I looked at LinkedIn, I found that I was connected to 8,900 people in the games and computing industries and am followed by another 8,640. Reach into industry is important if you are expecting to actually get a position in the games industry upon graduation. You might also do a Google search on the key personnel to see how they have appeared in news and press releases.

That is how you get a position in the games industry: education, education, and lots of game building. Have fun in your pursuit of that dream!

ACKNOWLEDGMENTS

When one establishes a games program, there are many people who contribute and help out. This list is not comprehensive. Gerard Medioni, chair of the USC Department of Computer Science, let me create the syllabi for 16 new courses and plan to establish the program. Chris Swain and Victor Lacour were instrumental in making the initial offerings of the advanced games projects course outstanding. Scott Easley, St John Colon, and Laird Malamed helped cast the AGP into something that will live forever at USC. All of my past students helped make the games program professional; great; and, most importantly, fun!

REFERENCES

1. "What is the size of Unreal Engine's source code currently?" Quora.com. https://www.quora.com/What-is-the-size-of-Unreal-Engines-source-code-currently (Accessed: Mar. 14, 2022).
2. [M. Zyda, "Educating the next generation of game developers," *Computer*, vol. 39, no. 6, pp. 30–34, Jun. 2006, doi: 10.1109/MC.2006.197.

MICHAEL ZYDA is the founding director of the Computer Science Games Program and a professor of engineering practice in the Department of Computer Science, University of Southern California, Los Angeles, California, 90089, USA. Contact him at zyda@mikezyda.com.

Conference Calendar

IEEE Computer Society conferences are valuable forums for learning on broad and dynamically shifting topics from within the computing profession. With over 200 conferences featuring leading experts and thought leaders, we have an event that is right for you. Questions? Contact conferences@computer.org.

AUGUST

7 August

- IRI (IEEE Int'l Conf. on Information Reuse and Integration for Data Science), San Jose, USA
- MIPR (IEEE Int'l Conf. on Multimedia Information Processing and Retrieval), San Jose, USA

19 August

- Blockchain (IEEE Int'l Conf. on Blockchain), Copenhagen, Denmark
- Cybermatics (IEEE Congress on Cybermatics), Copenhagen, Denmark

21 August

- HOTI (IEEE Symposium on High-Performance Interconnects), virtual
- RTCSA (IEEE Int'l Conf. on Embedded and Real-Time Computing Systems and Applications), Sokcho, South Korea

23 August

- DSC (IEEE Int'l Conf. on Data Science in Cyberspace), Jinan, China

25 Aug

- HCS (IEEE Hot Chips Symposium), Stanford, USA

27 Aug

- SustainTech (IEEE SustainTech Expo: Technology Solutions for a Sustainable Future), San Diego, USA

SEPTEMBER

2 September

- VL/HCC (IEEE Symposium on Visual Languages and Human-Centric Computing), Liverpool, UK

15 September

- IISWC (IEEE Int'l Symposium on Workload Characterization), Vancouver, Canada
- QCE (IEEE Int'l Conf. on Quantum Computing and Eng.), Montreal, Canada

16 September

- ACSOS (IEEE Int'l Conf. on Autonomic Computing and Self-Organizing Systems), Aarhus, Denmark
- e-Science (IEEE Int'l Conf. on e-Science), Osaka, Japan

23 September

- MASS (IEEE Int'l Conf. on Mobile Ad-Hoc and Smart Systems), Seoul, South Korea

24 September

- CLUSTER (IEEE Int'l Conf. on Cluster Computing), Kobe, Japan
- IC2E (2024 IEEE Int'l Conf. on Cloud Eng.), Paphos, Cyprus

OCTOBER

6 October

- ICSME (IEEE Int'l Conf. on Software Maintenance and Evolution), Flagstaff, USA
- VISSOFT (IEEE Working Conf. on Software Visualization), Flagstaff, USA

7 October

- SCAM (IEEE Int'l Conf. on Source Code Analysis and Manipulation), Flagstaff, USA
- SecDev (IEEE Secure Development Conf.), Pittsburgh, USA

8 October

- DFT (IEEE Int'l Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems), Didcot, United Kingdom
- LCN (IEEE Conf. on Local Computer Networks), Normandy, France

10 October

- ICPADS (IEEE Int'l Conf. on Parallel and Distributed Systems), Belgrade, Serbia

11 October

- ICEBE (IEEE Int'l Conf. on e-Business Eng.), Shanghai, China

13 October

- TopoInVis (IEEE Topological Data Analysis and Visualization), St. Pete Beach, USA
- VIS (IEEE Visualization and Visual Analytics), St. Pete Beach, USA

14 October

- VDS (IEEE Visualization in Data Science), St. Pete Beach, USA

 2469-7087/24 © 2024 IEEE

20 October
- FOCS (IEEE Annual Symposium on Foundations of Computer Science), Chicago, USA

21 October
- ISMAR (IEEE Int'l Symposium on Mixed and Augmented Reality), Bellevue, Washington, USA

28 October
- CIC (IEEE Int'l Conf. on Collaboration and Internet Computing), Washington, DC, USA
- CogMI (IEEE Int'l Conf. on Cognitive Machine Intelligence), Washington, DC, USA
- ICNP (IEEE Int'l Conf. on Network Protocols), Charleroi, Belgium
- ISSRE (IEEE Int'l Symposium on Software Reliability Eng.), Tsukuba, Japan
- TPS-ISA (IEEE Int'l Conf. on Trust, Privacy and Security in Intelligent Systems, and Applications), Washington, DC, USA

30 October
- BDCloud (IEEE Int'l Conf. on Big Data and Cloud Computing), Kaifeng, China
- ISPA (IEEE Int'l Symposium on Parallel and Distributed Processing with Applications), Kaifeng, China
- SpaCCS (IEEE Int'l Conf. on Security, Privacy, Anonymity in Computation and Communication and Storage), Kaifeng, China
- SustainCom (IEEE Int'l Conf. on Sustainable Computing and Communications), Kaifeng, China

NOVEMBER

3 November
- ICCD (IEEE Int'l Conf. on Computer Design), Milan, Italy

13 November
- PRDC (IEEE Pacific Rim Int'l Symposium on Dependable Computing), Osaka, Japan

14 November
- SmartIoT (IEEE Int'l Conf. on Smart Internet of Things), Shenzhen, China

15 November
- MedAI (IEEE Int'l Conf. on Medical Artificial Intelligence), Chongqing, China

22 November
- IPCCC (IEEE Int'l Performance, Computing, and Communications Conf.), Orlando, USA

27 November
- BIBE (IEEE Int'l Conf. on Bioinformatics and Bioengineering), Kragujevac, Serbia

DECEMBER

4 December
- ICA (IEEE Int'l Conf. on Agents), Wollongong, Australia

9 December
- ICDM (IEEE Int'l Conf. on Data Mining), Abu Dhabi, United Arab Emirate

10 December
- RTSS (IEEE Real-Time Systems Symposium), York, UK

15 December
- BigData (IEEE Int'l Conf. on Big Data), Washington, District of Columbia, USA

18 December
- HiPC (IEEE Int'l Conf. on High Performance Computing, Data, and Analytics), Bangalore, India

19 December
- ESAI (Int'l Conf. on Embedded Systems and Artificial Intelligence), Fez, Morocco

2025

JANUARY

27 January
- AIxVR (IEEE Int'l Conf. on Artificial Intelligence and eXtended and Virtual Reality), Lisbon, Portugal