



Autonomous Tools in System Design: Reflective Practice in Ubisoft's *Ghost Recon Wildlands* Project

Stefan Seidel, University of Liechtenstein

Nicholas Berente, University of Notre Dame

Benoît Martinez, Ubisoft Paris

Aron Lindberg, Stevens Institute of Technology

Kalle Lyytinen, Case Western Reserve University

Jeffrey V. Nickerson, Stevens Institute of Technology

Ubisoft's game designers successfully used autonomous tools to develop an innovative virtual world. The authors discuss the reflective practices underlying this success and how autonomous tools enable more complex system design.

Designers increasingly use autonomous tools while executing design tasks.¹⁻³ These tools employ artificial intelligence (AI) and related algorithmic methods including machine learning, pattern recognition, meta-heuristics, and evolutionary algorithms to generate artifacts that cannot be created by most—or any—humans. The level of

autonomy varies, from tools that generate results from the input of designers that cannot be fully anticipated by those designers, to more granular tools that improve human performance and learn as they go. Generally, autonomous tools let designers explore design spaces across a wide range of parameters, enabling them to produce complex and complete outcomes. Such tools

have given rise to various novel design approaches such as procedural generation,^{4,5} procedural modeling,⁶ and computational creativity.⁷

Autonomous tools reshape design processes. Tasks formerly conducted by human designers are delegated to tools that, in turn, redefine those tasks. The process is iterative. To understand how to use these tools effectively, designers constantly evaluate the outcomes that the tools generate. The designers further develop and adjust the tools, and make new decisions about them. In this way, autonomous tools invite designers to steer design processes mindfully in their effort to generate novel and useful outcomes—what Donald Schön calls *reflective practice*.⁸

In this article, we examine a set of reflective practices that emerged during the creation of a new, large-scale action adventure game called *Tom Clancy's Ghost Recon Wildlands* by Ubisoft, a major video game developer and publishing company. Throughout the development process, Ubisoft's designers used autonomous tools extensively to generate significant portions of the game's unusually large world. The tools, combined with human ingenuity and craftsmanship, created detailed, customized, and innovative game features, allowing the developers to explore wider solution spaces. We conclude by addressing how organizations and designers can embrace new kinds of human-machine interactions offered by autonomous tools.


THE PROMISE OF AUTONOMOUS TOOLS: SCALE, ITERATION, CREATIVITY

Designers leverage autonomous tools to achieve benefits that accrue from

scale, iteration, and creativity afforded by such tools.

Autonomous tools can execute large volumes of repetitive tasks and *scale* easily when design tasks increase in size and scope. Contemporary video games need to provide large amounts of content within open world settings, across which the game is played. Such large-scale virtual environments cannot be developed without autonomous tools, given the resource constraints under which commercial game developers operate. Autonomous tools render development processes more efficient and encourage the exploration of larger design spaces, which in turns leads to

designers can shift their focus toward the *creativity* needed for aspects of the design that demand human judgment and sensibility as well as customized solutions. As autonomous tools generate additional content by actively participating in the design process, designers can choose different evaluation criteria and introduce additional dimensions to the design—often resulting in surprisingly creative outcomes. In video games, for instance, they can create more immersive user experiences by aligning narratives, visuals, environmental settings, and the overall tone of a particular scene or situation. Designers can thus



DESIGNERS LEVERAGE AUTONOMOUS TOOLS TO ACHIEVE BENEFITS THAT ACCRUE FROM SCALE, ITERATION, AND CREATIVITY AFFORDED BY SUCH TOOLS.

the generation of new kinds of design outcomes at unprecedented scales.

Autonomous tools also allow for an increased number of *iterations*. They generate and test a larger scope of design alternatives than is otherwise possible. Iteration forms an essential feature in designs where creativity matters and multiple potential solutions to a given problem need to be explored. Designers use autonomous tools to experiment with new versions of the game, test numerous combinations of suggested game elements, and explore varying aspects of the design space.

Finally, by off-loading repetitive tasks onto autonomous tools,

focus their attention on those facets of the game where their creativity will achieve the highest impact with regard to developing richer user experiences, such as detailing the settings where players will spend large proportions of their time.

These three characteristics are complementary and mutually reinforcing, promoting superior design outcomes. Such benefits, however, are not guaranteed. To be successful using autonomous tools, designers must revamp their understanding of system design and revisit assumptions about essential capabilities. They also must change how they organize and coordinate design projects.

DESIGNING GHOST RECON WILDLANDS WITH AUTONOMOUS TOOLS

In 2012, a small team at Ubisoft Paris was tasked with developing a new video game for the highly successful Tom Clancy's *Ghost Recon* tactical shooter franchise. *Ghost Recon Wildlands* was to feature an unusually large game world, modeled on real-world Bolivia, within which the user could freely move and explore. The world-building design team eventually grew to more than 50 people, but this was still relatively small compared to the industry practice for this type of game. It was the first major Ubisoft project to extensively leverage autonomous tools.

Designers handcrafted detailed elements in the game world while algorithms procedurally generated most of the background content. This arrangement permitted the team to focus on creative tasks in lieu of mechanical, repetitive tasks. The tools would, for instance, generate large amounts of detailed terrain; then the designers would modify the terrain further and generate additional detail for each setting. The *Ghost Recon Wildlands* designers were not excluded from the creative design process, but their role changed. The design effort now included understanding how tools could be used in terms of what parameter settings were available, and experimenting with the tools until a satisfactory outcome was achieved. These activities had to be integrated with traditional manual design activities, which evolved in parallel because many key areas of the game world continued to be designed manually.

The development process integrated autonomous tools and human craftsmanship, and called for selecting

and developing appropriate tools and models that would align with the game concept. To this end the design team built an infrastructure model that included systems for terrain erosion, roads and railway lines, vegetation, placement of rocks, rivers and streams, cave networks, villages, traffic signs, and power lines—each associated with a specific set of modeling rules. These tools were built primarily with the popular 3D modeling software Houdini,⁹ complemented by several proprietary in-house tools. The autonomous tools used a variety of intelligent methods such as pathfinding, fluid dynamics, and packing algorithms.

A key challenge in this step was figuring out how to manage interdependencies among various tools. 3D games involve a multitude of interconnected game elements—visual elements including terrain, vegetation, roads, and so on but also logic elements such as probabilistic, conditional trajectories for AI vehicles or non-playable characters (NPCs), among others. To address this challenge, Ubisoft introduced *subtractive workflow*, where those parts of the game world not affected by a change were automatically excluded from design activity. Designers could focus on selected elements and then, overnight, the automated tools would collect information about related assets such as rocks and vegetation, check for interdependencies, exclude irrelevant parts, and update the changes to render the entire game world anew.

The autonomous generation of the road network within the game world provides a good illustration of how autonomous tools were used to design content essential for playing *Ghost Recon Wildlands*, but that did not require the creativity of human

designers. Having the road network built by an autonomous tool freed up development resources to focus on other parts of the game, such as villages connected by those roads. At the same time, roads had to meet criteria such as not exceeding a certain steepness. After some experimentation, the design team used a tool to compute the trajectories of roads and associated terraforming. Figure 1 shows how the terrain evolved as a road network was added algorithmically based on a select set of defined waypoints across the terrain. Similar autonomous processes were used to generate other basic game elements such as vegetation, as well as more complex elements such as villages.

The illustration highlights how the design team integrated manual and automated design, and balanced automation with design craftsmanship. It also highlights how scale, iteration, and creativity are interconnected characteristics of using autonomous tools—such tools allowed for the building of a large network of major and minor roads (scale), but this generation required the designers to run multiple tests and modify the algorithms used (iteration), while the largely automated process let the designers focus on creating immersive user experiences elsewhere (creativity). Generally, the designers handcrafted those elements of the game that involved intensive user interaction and immersion. Repetitive and mechanical tasks, such as the generation of generic content in backgrounds, were automated. In such a hybrid model of design, tools and designers jointly generate the outcome for a given problem space. The success of this process depends on implementing new, reflective design practices that can support each specific



FIGURE 1. Algorithmic generation of a road network based on a set of defined waypoints in *Tom Clancy's Ghost Recon Wildlands*. Similar autonomous processes were used to generate other basic game elements such as vegetation, as well as more complex elements such as villages. (Image: Ubisoft)

task that emerges within the hybridized form of design.

REFLECTIVE DESIGN PRACTICE

Great designs involve a form of reflective practice⁸ where the designers engage in an iterative conversation between themselves and the different ways in which they represent potential solutions. Designers continually

reflect on their experiences, advance different formulations of the problem, and explore alternative avenues to solve them through novel designs. This reflective practice takes different forms when using autonomous tools, requiring the confluence of

- › choosing and developing tools,
- › using tools to iteratively develop design outcomes, and

- › constantly reflecting on both the outcomes of the tool and the capabilities and adequacy of the underlying infrastructure of tools.

Autonomous tools act, at least partially, independently of the designer, and reflection occurs in relation to both the designer's activities and the tools used—for example, reflectively

TABLE 1. Reflective design practice for succeeding with autonomous tools.

Design activities	Description	Illustration from <i>Ghost Recon Wildlands</i>
Tool design and orchestration	Design teams shift the focus toward developing and integrating autonomous tools to contribute synergistically to the design product.	Autonomous tools generated different elements of the game world and had to be orchestrated. A “subtractive workflow” allowed individual designers to focus on their individual tasks while providing for the integration of various tools as well as rendering the whole game world and associated interdependencies.
Parameterization, execution, and evaluation	Designers must understand how to specify parameter inputs, run the tool, evaluate the output, and make decisions on subsequent parameter settings and experiments based on reflection and learning.	Designers working with an autonomous tool for generating vegetation had to understand how input parameters of different kinds of shrubbery and foliage match with the surrounding terrain.
Reflection and learning	Two basic types of reflection address (a) use of the tool and (b) tool design and orchestration: Designers reflect on the results of the tool in-design—this process can result in changes to the design process, for instance, in the levels of various parameters set in the autonomous tool. Designers learn about the tool design, including the assumptions that are embedded in the tool—this process can result in changes to the tool.	Designers learned how to use a variety of tools, for instance, to generate vegetation (what combinations of parameters lead to successful outcomes). Tools were adjusted and further developed in accordance with what designers expected.

constructing local theories of the tools’ behavior. Table 1 summarizes key activities that occur in reflective design practice using autonomous tools, exemplified by illustrations from *Ghost Recon Wildlands*.

The use of autonomous tools in system design involves a shift from manual craftsmanship toward *tool design and orchestration*: selecting, designing, configuring, and administrating these tools. Designers who can use, evaluate, and create such tools are in high demand. This does not necessarily mean that designers must understand a tool’s inner workings (for example, the machine learning algorithms), but they have to understand what the tool does and what input is transformed into what output. As tools autonomously develop design elements that are then combined with manually generated elements, multiple manual

and autonomous tools must be orchestrated. This orchestration involves conceptual aspects such as assessing interdependencies among design elements, as well as technical aspects related to input and output formats of content generated by the tools. At Ubisoft, a small team developed and orchestrated an entire infrastructure of tools that allowed the team to build a complex game.

As the focus shifts away from manual design work, designers must excel in using autonomous tools. Specifically, they must understand how varying parameter settings lead to different results, and how those results can be integrated with other elements of the final product. *Parameterization, execution, and evaluation* are interrelated activities that replace manual craftsmanship for separate design tasks. Designers can influence the design

process through setting parameters, instead of handcrafting elements of the design artifact, and can experiment quickly with multiple alternative settings. This represents an alternative approach to traditional, manual design tasks. Experimentation involves testing how different parameter settings impact the design outcome, and therefore the designer needs new skills to evaluate intermediate outcomes across the design process. Traditionally, designers evaluate the outcomes of their own manual activity; now, they must evaluate what the tool has generated. In the case of *Ghost Recon Wildlands*, the designers evaluated multiple game elements such as road networks, but such road networks were then combined with other game elements, and the emergent game world was also evaluated holistically by each designer. Moreover, given the

large number of possible variations, automated evaluation that eliminates infeasible designs and promotes feasible and novel designs streamlines the evaluation task.

Such iterative parameterization, experimentation, and evaluation requires designers to embark on *reflection and learning* that occurs at two levels: learning about the tool in-design and learning about the tool design and orchestration. Based on the evaluation of the results of using autonomous tools, designers learn about the process of using the tools, thereby gaining an improved understanding of what parameter settings lead to what results, and how these results align (or not) with other design elements. While learning about the tool in-design can solve specific design issues through making specific design decisions, and help individual designers as well as the design team to iteratively develop solutions, there are occasions where the design team is confronted with novel constraints—that is, the tools cannot realize the design vision. The team can then change the tool design, and revise the assumptions embedded in the tool. In the case of *Ghost Recon Wildlands*, the design team started with no autonomous tools, but over time they developed an entire infrastructure of such tools including tools not formerly autonomous but enhanced with autonomous capabilities during development. This resulted in a new form of hybrid human-machine learning¹⁰ that requires tools to continually be aligned and realigned with the design team’s mental models.

Figure 2 illustrates the confluence of key activities in reflective design practice when autonomous tools are used. First, designers chose, develop,

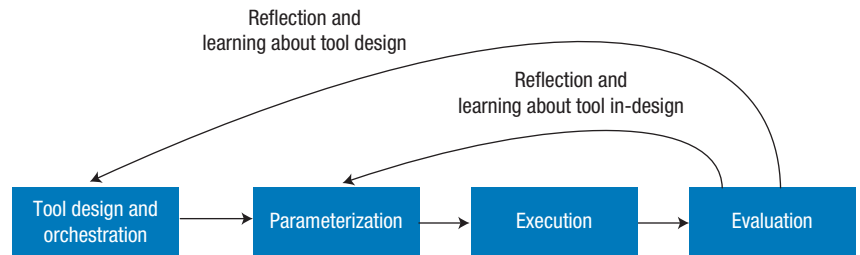


FIGURE 2. Activities of reflective practice in designing with autonomous tools. The confluence of these activities allows for large-scale projects as the tools generate content through multiple, parallel experiments, eventually freeing up designers to focus on creativity-intensive work.

and orchestrate tools. These tools then need to be parameterized to run multiple experiments. As designers reflect on use of the tools, they may change how they use them (what parameters they set) or even go back to tool design and orchestration. Together, these activities allow for large-scale projects as the tools generate content through multiple, parallel experiments, eventually freeing up designers to focus on creativity-intensive work.

BROADER IMPLICATIONS—WHO WINS AND WHO LOSES?

The interaction of human craftsmanship and autonomous tools offers novel opportunities in terms of scale, iteration, and creativity for design with important implications for both designers and the users of the resulting systems. Designers can explore larger spaces and generate solutions that would not have been possible without the use of autonomous tools. Moreover, they can do so at a rapid pace. Users can enjoy designs of unprecedented scale. In the case of video games, they can immerse themselves in worlds characterized by vastness (enabled by the generation of content through

autonomous tools) and a rich user experience (enabled by the increased creativity of human designers).

These results do not come free. They require changes to how designers and design teams approach design, and therefore require that designers build new capabilities. Further, a good deal of activity can now be carried out by autonomous tools and remains largely black-boxed. In this setting human designers still play a primary, leading role. This hybrid model of human-machine system design also comes with several challenges related to

- › finding an appropriate balance between automation and craftsmanship,
- › the training and capabilities of designers, and
- › the organization of design work.

To be successful with autonomous tools, designers and design teams must identify what types of work can be effectively automated, as well as whether the results meet expectations set at the outset of the design process. It is likely that design activities where intuition, creativity, and unexpected associations are crucial will be carried

out by human designers, while repetitive activities characterized by scant creativity are increasingly automated. Usually the few elements most important for user experience and immersion must be handcrafted while the generation of peripheral content can be automated. Design teams must continuously make decisions about what elements need to be handcrafted and what elements can be automatically generated. These decisions have far-reaching consequences for both the design of autonomous tools as well as the final outcome of the design processes themselves.

The role of designers—and thus their training and capabilities—is also changing. Designers envisioning artifacts must interact with autonomous tools in ways that help them realize increasingly ambitious design visions. In some cases the designers may become further removed from manipulating actual artifacts, but this also enables them to create new artifacts. Designers need the skills to evaluate complex results generated by a tool to understand how the outcome relates to the tool's input parameters and setup. Organizations are challenged to provide ample opportunities to adopt and develop new design practices, and those that appreciate the careful balance between human craftsmanship and technology innovation will succeed. At the same time, universities must attend to these forms of design across various fields, including software development and many other design disciplines. The amount of design work carried out by humans in cooperation with autonomous tools will only increase.

Finally, use of autonomous tools requires rethinking how design activities are organized. Successful design

teams will shift attention toward building and orchestrating tools, but this process must be aligned with the primary design process, leading to new learning processes centered on the tools. New types of iterations between building tools and carrying out proper design work will emerge. Companies that do not use autonomous tools will lose out to better-prepared competitors, while organizations that blindly trust tools and do not continuously reevaluate tool interactions with human designers will fail. Organizations must not only build strong tools but also encourage exploratory learning by designers with regard to tool design and how to use tools in-design.


THE ROAD AHEAD: MACHINE LEARNING AND INCREASED AUTONOMY

Autonomous tools provide unprecedented opportunities for creative problem solving in multiple design domains ranging from game design, architecture, and mechanical engineering to semiconductor chip design. Succeeding with autonomous tools in system design calls for new competencies in building and orchestrating tools and a reorganization of design processes, as well as a good understanding of where autonomous tools can be used and where they should not be used. In sum, the integration of autonomous tools into design processes leads to questioning long-held assumptions about design as a process of manual craftsmanship.¹¹ Future designers must rethink their role in the design process that used to center on them but is now increasingly centered on the use of autonomous tools.

While important elements of a design artifact can be generated using autonomous tools, it is clear that the

role of the human designer will not disappear, but rather is changing. Elements of manual craftsmanship remain important as designers can now focus on high-impact aspects of design that require intuitive capacities such as how to shape and understand user experiences.

The autonomy of tools can vary and largely depends on the tool's ability to learn. In the *Ghost Recon Wildlands* project, tools were mainly used to generate content such as using a pathfinding algorithm to create a road network. The next step is to make use of machine learning as an approach to learn from existing datasets, thereby increasing the tools' autonomy. Machine learning will open new perspectives for system design, as the results are expected to compete with the quality of content that has been created manually or procedurally with current tools. Interface designs already use machine learning extensively,¹² and machine learning has also been used to generate new solutions to play video games.¹³ As humans and machines interact in design they combine multiple forms of human learning, machine learning, and joint human-machine learning, leading to an entirely new breed of design processes.

It will be exciting to follow how design professions embrace new forms of digitization that come from the use of autonomous tools. Neither sticking to long-held assumptions about design as human craftsmanship nor expecting autonomous tools to solve every design problem is warranted. Rather, we must reflect deeply and deliberately on the new opportunities that autonomous tools provide across design professions and tasks. 

ACKNOWLEDGMENTS

This material is based upon work partially supported by the National Science Foundation (Grants IIS-1422066, CCF-1442840, IIS-1717473, and IIS-1745463).

REFERENCES

1. J. Ambach and A. Reppenning, "Participatory Theater: Interacting with Autonomous Tools for Creative Applications," *Knowledge-Based Systems*, vol. 9, no. 6, 1996, pp. 351–358.
2. G. Davenport and B. Bradley, "The Care and Feeding of Users," *IEEE MultiMedia*, vol. 4, no. 1, 1997, pp. 8–11.
3. D. Lozowski, "Advancing with Automation," *Chemical Eng.*, vol. 123, no. 3, 2016, p. 5.
4. M. Hendriks et al., "Procedural Content Generation for Games: A Survey," *ACM Trans. Multimedia Computing, Comm., and Applications*, vol. 9, no. 1, 2013; doi:10.1145/2422956.2422957.
5. D. Ashlock and C. McGuinness, "Landscape Automata for Search Based Procedural Content Generation," *Proc. 2013 IEEE Conf. Computational Intelligence in Games (CIG 13)*, 2013; doi:10.1109/CIG.2013.6633619.
6. P. Müller et al., "Procedural Modeling of Buildings," *ACM Trans. Graphics*, vol. 25, no. 3, 2006, pp. 614–623.
7. A. Liapis, G.N. Yannakakis, and J. Togelius, "Computational Game Creativity," *Proc. 5th Int'l Conf. Computational Creativity (ICCC 14)*, 2014, pp. 46–53.
8. D.A. Schon, *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, 1984.
9. K. Xu and D. Campeanu, "Houdini Engine: Evolution towards a Procedural Pipeline," *Proc. 4th Symp. Digital Production (DigiPro 14)*, 2014, pp. 13–18.
10. S. Seidel et al., "Autonomous Tools

ABOUT THE AUTHORS

STEFAN SEIDEL is a professor and Chair of Information Systems and Innovation at the Institute of Information Systems at the University of Liechtenstein. His research explores the role of digital technologies in creating organizational, social, and environmental innovation and change. Contact him at stefan.seidel@uni.li.

NICHOLAS BERENTE is an associate professor at the University of Notre Dame's Mendoza College of Business. He researches digital innovation, organizational change, and cyberinfrastructure, and is associate editor of *MIS Quarterly*. Contact him at nberente@nd.edu.

BENOÎT MARTINEZ is a lead artist and technical art director at Ubisoft Paris, where he creates large-scale environments and focuses on innovative techniques, tools, and production strategies, including procedural approaches and machine learning. Contact him at benoit.martinez@ubisoft.com.

ARON LINDBERG is an assistant professor of information systems at the School of Business, Stevens Institute of Technology. He primarily studies complex design processes, often using a combination of qualitative and computational methods. Contact him at alindber@stevens.edu.

KALLE LYTTINEN is a Distinguished University Professor and the Iris S. Wolstein Professor of Management Design at Case Western Reserve University. An Association of Information Systems Fellow, he conducts research on digital innovation, design work, requirements in large-scale systems, diffusion and assimilation of innovations, and emergence digital infrastructures. Contact him at kjl13@case.edu.

JEFFREY V. NICKERSON is a professor and Associate Dean of Research at the School of Business, Stevens Institute of Technology. His research focuses on collective creativity, the way knowledge emerges from the interactions of people and machines. Contact him at jnickerson@stevens.edu.

- & Design Work: A Triple-Look Approach to Human-Machine Learning," to be published in *Comm. ACM*.
11. R. Sennett, *The Craftsman*, Allen Lane, 2008.
 12. M.E. Yumer et al., "Procedural Modeling Using Autoencoder Networks,"

Proc. 28th Ann. ACM Symp. User Interface Software & Technology (UIST 15), 2015, pp. 109–118.

13. V. Mnih et al., "Human-Level Control through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, 2015, pp. 529–533.