Ethical design of social simulations

Marco Almada, Romis Attux

School of Electrical and Computer Engineering
University of Campinas (Unicamp)

marco.almada@usp.br, attux@dca.fee.unicamp.br

Abstract. *Computer simulations of social phenomena are used in academic and non-academic contexts as tools for decision-aiding. By combining social scientific knowledge, computational power, and large data sets, those systems may be involved in decisions that affect people who are not even aware of the existence of a simulation. This paper reviews topics from the philosophy of technology and computer ethics literatures in order to identify salient ethical issues related to computer simulations and the steps on the software design cycle on which they can be addressed, offering a checklist that can guide how the ethical concerns of stakeholders are listened to and incorporated into the proposed simulation system.*

1. Introduction

Computer simulations of social phenomena play many roles at various aspects of modern society: they can be used in social-scientific studies [Elsenbroich and Gilbert 2014], as forms of entertainment [Koabel 2017], and within social credit systems [Yu *et al*. 2015] such as those currently in adoption in China [Hvistendahl 2018], among other uses. Since those applications play a growing role on how individuals and organizations make decisions, there are growing concerns about the transparency of the data and algorithms employed on simulations [e.g., Pasquale 2015], which may introduce or reinforce discriminatory patterns against socially vulnerable groups [O'Neil 2017; Eubanks 2018], issues that have been addressed both through technical approaches [ACM Public Policy Council 2017] and direct legal regulation [Ohm and Reid 2016].

To better understand how computer simulations can impact social life, this paper draws from a post-phenomenological philosophical framework [Verbeek 2005] to explain the different modes of interaction between computer simulations and individuals or groups. This analysis works as a start point for exploring ethical concerns such as biases and lack of accountability [O'Neil 2017], which then drive a targeted, non-systematic review of the simulation design literature. Finally, that review is then used as a basis for a checklist of measures to identify the relevant stakeholders and address their ethical concerns within the simulation development cycle.

This paper frames some of the current ethical issues related to computer simulations in terms of how they present themselves during the software development process. From this social and philosophical basis, simulation designers can then address ethical issues as additional requirements, to be treated according to best practices in software engineering. The proposed checklist identifies actionable points in which ethical concerns can be addressed, providing a starting ground that can be refined from the presented conceptual framework and the concrete demands of simulation projects.

2. Prior work

The object of the present discussion is how software developers can address ethical issues that arise when building computer simulations that describe or interact with social phenomena. Since those simulations have a wide range of applications, some domain-specific concerns will almost always be present, and an ethical framework for simulation must be adaptable enough to handle such cases. As a consequence, an approach that establish generic standards of ethical professional behavior will not suffice to cover entirely the ethical demands of simulation design, as Floridi and Sanders [2002] point out, and this section explores possible alternatives.

2.1. Computer simulations and social phenomena

Cioffi-Revilla [2014] presents a treatment of simulation-specific ethics, extending the *Truth-Beauty-Justice* framework proposed for assessment of formal models in social science. While beauty, defined in terms of formal aesthetics of modelling and programming, may have ethical consequences — for example, a minimalist simulation may be ethically desirable due to its reduced software and hardware demands, which in turn enable more people to run the simulation and reduce its ecological footprint —, most of the ethical concerns present in simulation design can be associated to truth and justice concerns. The *truth* factor in this ethical requirement can be associated with the epistemic requirements for using a computer simulation, that is, the conditions that must be met before one can trust that the simulation produces reliable results, such as accuracy; *justice*, in contrast, must be understood as a outward-looking target that relates the simulation and its modes of use to desirable social states.

The general topic of justice is object of a vast literature that could not be adequately surveyed within the scope of this paper; however, as Sen [2009] points out, a comparative approach can help ameliorate patently unjust situations even in the absence of a more general conception of a just society. In the context of computer ethics, one such approach can be seen at the end of Floridi and Sanders [2002]'s review of computer ethics, where the authors suggest that information-related notions such as information objects and entropy can be used to identify what are the relevant constraints in the environment in which simulations are deployed, allowing for a more apt description of the ethical demands related to a simulation in a given context.

An environmental approach becomes even more suitable when one takes into account that using computer simulations in ethical ways demands effort from a wide set of stakeholders: not only those responsible by programming the system itself, but also the model designers, those who actually use the models as part of their knowledge-acquisition or decision-making processes, among others. Even when they do not interact directly with the computer simulations, stakeholders can have their interactions with the world shaped by them; O'Neil [2017] cites the example of the 2016 US presidential elections, in which Clinton's decision not to focus on campaigning at the states of Michigan and Wisconsin, resulting in electoral losses at both states, was partially driven by the results of a simulation algorithm named Ada.

Verbeek [2005] presents a model to understand the ways in which technological artifacts can shape human interaction with other individuals and the world in general,

some of which which can be applied to computer simulations. A first mode of interaction between individuals and simulations is characterized by *hermeneutic* relations, in which simulations provide a representation of the world that must be interpreted by the user, as is the case when a hedge fund manager uses a Monte Carlo model as a proxy for deciding whether they should pursue or not a given investment. If, instead, an individual interacts with a simulation directly and not as a proxy — for example, when playing a computer simulation game —, the ensuing *alterity* relation emphasizes the (mostly) autonomous behavior of the simulational artifact.

Both forms of interaction, in Verbeek [2005]'s framework, are different from *background* relations, in which the presence of the artifact is barely noticed — or not at all — during normal operation: a bank customer's loan application may be conceded or denied based on simulation outputs and credit score values, but that person will hardly, if ever, be exposed do the actual simulation. As the growing availability of data allows the use of complex simulations in ever more important roles in business and government processes, individuals and communities will be subject to the economical, political, legal, and social consequences of decisions taken based on models that they do not know about and cannot control, making it more difficult to prevent unfair outcomes or to reverse them.

Lack of access to computer simulations becomes an even greater issue when one takes into account their *black box* nature: corporate practices, characteristics of algorithms and systems, and technical illiteracy among stakeholders [Burrell 2016] can prevent stakeholders from identifying or fixing unfair or otherwise inaccurate outputs that may arise from inadequate input data or other issues during simulation design and usage. This lack of understanding about computer simulations is compounded by what Pasquale [2015] termed a *black box society*: a complex social context in which interaction or even knowledge about simulations and related systems is mediated by specific rules, such as non-disclosure agreements and end-user licence agreements.

Due to those multiple sorts of opacity, non-user stakeholders rarely, if ever, have access to all the information needed for weighing the involved ethical factors, and even direct users might not be fully aware of the implications of adopting a given simulation model. Therefore, model and software designers must take an additional burden of understanding the actual consequences of placing computer simulations in relation with people, especially when those interactions happen in the background of the lives of individuals and groups, without their informed consent or knowledge.

2.2. Simulations as social systems

Discussions on the ethical aspects of computer simulation are nowadays usually happen within a more general debate on the use of computer-driven decision-making and *big data* [e.g., O'Neil 2017; Pasquale 2015; Ohm and Reid 2016; Eubanks 2018]. While those debates are usually framed around issues such as algorithmic fairness or algorithmic discrimination, they actually refer to *computer systems* which implement those algorithms and, by establishing relations between individuals and their environment such as those described in the previous subsection, affect not only their users but also third parties that might not even be aware of the system's existence.

Simulations can produce social effects by shaping modes of interpersonal interaction; they may thus lead to ethical violations if their design or use somehow conflicts with existing ethical duties, such as the respect for the privacy of its users. Those ethical hazards will depend on the specific nature of the relation mediated by a given simulation in a given context: for instance, simulations can lead to unethical results in a hermeneutical role if they fail to take into account relevant modeling aspects, while the background effect of simulation-aided decisions may force third parties into avoidable risks or unduly constrict their choices.

An example of ethical hazards induced by computer simulations can be seen from the risk models used by financial institutions prior to the 2008 economic crisis [Mackenzie and Spears 2014]. Those models, heavily reliant on the results of Monte Carlo methods and other simulations, guided investment decisions on banks; their normal operation brought significant profits, but failure to take into account possible modes of failure misled not only the direct users of those simulations — who then provided inaccurate advice to their internal and external customers — but also to the homeowners who held subprime mortgages and, ultimately, to the global economy.

O'Neil [2017] describes algorithms such as the aforementioned financial simulations or the ones used in prisoner parole evaluation as *weapons of math destruction*: computer systems that negatively impact the lives of millions of people without giving voice to the interests of those harmed. Those systems, by O'Neil [2017], are marked by three traits: they *damage* people's lives — by depriving them of options, as is the case on biased parole systems, or resources such as money —, operate at a *scale* that reaches thousands, if not more, of people — leading to feedback loops and emergent effects — and are *opaque*, in the senses discussed by Burrell [2016], preventing the parties harmed cannot understand what happens within the system in order to eliminate the negative outcomes.

As simulations are understood as intrinsically opaque models [Di Paolo *et al*. 2000], their design and use must be especially aware of how, deployed at scale in real contexts, simulation outputs might be used in ways that harms the rights of direct and background stakeholders. Still, as Doshi-Velez and Kortz [2017] aptly point out, the demands for algorithmic transparency are rarely absolute, as different contexts have different demands on what counts as a valid explanation of simulation outputs. Careful simulation design might, then, provide simulations with socially acceptable levels of result explainability, which can then be used to mitigate or avoid the damages they cause to the lives and rights of those directly or indirectly affected by the simulation.

2.3. Techniques for simulation design

Since computer simulations can shape, directly or indirectly, the perceptions and behavior of individuals far removed from the actual models, their design should take into account how the simulations can affect those people. A starting point for this can be drawn from the ACM Public Policy Council [2017]'s guidelines on algorithm transparency, which propose that computer models, algorithms, data, and decisions should be well-validated, well-understood, and auditable; furthermore, all stakeholders should be aware of possible harms coming from the resulting systems, and should be able to seek redress for any adverse effects.

Textbooks on the design of social simulations [such as Elsenbroich and Gilbert 2014; Gilbert and Troitzsch 2005; De Marchi 2005] usually bypass the potential ethical issues on simulation design and use. An exception, already discussed, is Cioffi-Revilla [2014]'s treatment of the *Truth, Beauty, and Justice (TBJ)* framework, which establishes criteria that can shape the requirements of the software engineering processes that build a given computer simulation. Operationalizing such values, however, requires that the simulation designers identify and address potential ethical concerns related to possible stakeholders, a task that can benefit from the existing software engineering literature.

Searching Google Scholar for all scholarly works between 2011 and May 2018 containing the terms "computer simulation" and "ethics" returns about 14,900 results. Most of that literature does not focus on design issues; in fact, adding the term "software engineering" to the search reduces the result set to 485 works. Even contemporary reviews of computing ethics [such as Stahl *et al.* 2016] do not emphasize simulation-specific issues. As a consequence, this paper is built around a directed discussion of core references cited by that literature rather than a systematic review of the works found by a refined version of the described search.

From a designer perspective, Ören *et al*. [2002] propose a code of ethics for designers of computer simulations; among the commandments established by this code, it is possible to identify some ethical issues — such as respect for intellectual property and due credit, avoiding harms to humans and the environment, and disclosure of system assumptions, limitations, and conditions of applicability — which address the social concerns presented so far in the text. As designed, though, the code shows the strengths and limitations of what Floridi and Sanders [2002] termed a "professional ethics" approach to computer ethics: they provide clear guides to action, but fail to address the rights and positions of other stakeholders, e.g. in background interactions.

As simulations of social domains can shape how their users and third parties interact with the world, their design should take into account the *persuasive* role that simulations take in such interactions. Davis [2009] provides an overview and discussion of persuasive software design approaches, and ultimately proposes a combination of two frameworks to ensure that any persuasion happens in ways compatible with the beliefs and values held by stakeholders. By using the *Value Sensitive Design* (VSD) framework, a systems designer can combine conceptual, theoretical, and empirical investigations to identify what values are relevant to stakeholders and should therefore be preserved to the largest extent possible. This approach can be complemented, as suggested in the same work, by *Participatory Design* (PD) methods that actively involve stakeholders in the various stages of the design cycle. Simulation designers can then obtain a fuller picture that will allow them to address in the concrete case design values such as the TBJ framework, result trustworthiness, and a taxonomy of ignorance [Williamson 2010] that describes how absence or distortions of knowledge and uncertainty or inaccuracy on the results can affect the decisions that are based on a given simulation.

3. Ethical constraints and the simulation development process

Computer simulations of social phenomena draw heavily on the mainstream software engineering literature. A representative example can be seen in Siegfried [2014]'s

proposal of a framework for agent-based models, which is divided in seven stages: a *preliminary* stage that identifies the needs of relevant stakeholders is followed by the construction of a logical *problem definition* that addresses the previously identified requirements. From this formulation, the simulation developers then produce a conceptual model of the target system through a *target system analysis* that is used for system *formalization* and *implementation*. Through an *experimentation* process, the ensuing computational system is adjusted so as to produce actionable insights through specialized *interpretation*. Each of those stages can be mapped to one or more of the traditional software engineering activities of specifying, designing and implementing, validating, and evolving a system[Sommerville 2011], but this domain-specific frame emphasizes how simulations are embedded into a social context of use.

Before identifying what are the needs and demands of each relevant stakeholder, one must identify *who* they are. Prior knowledge about the relevant phenomenon can be used to obtain a first approach to the set of affected people, but even so it will be rarely possible to include all of them into an actual software design process. Thus, some form of sampling might be necessary. A fair sample of the stakeholder population must capture a diverse range of positions, and a way to accomplish that is to oversample those populations that domain-specific knowledge identify as particularly vulnerable to ethical hazards from the simulation. While such oversampling can bias the relevant-stakeholder pool, it does so in a way that can help simulation developers to hear from perspectives that would otherwise be ignored; requirement analysis techniques can then be employed to identify unexpected side effects that would otherwise only be found after deploying the simulation system.

From the separate impressions of the relevant stakeholders, the simulation designers will then build a formal definition of the problem that must be solved by the resulting system. To do so, they must reconcile the ethical (and other) preferences and values identified in the previous step. In some cases, those ethical requirements can act as *side constraints* to the ultimate simulation goals: a simulation built to evaluate the installation of hypermarkets at a highly religious region can still provide useful information even if it must take into account the fact that a sizeable fraction of the stakeholders find it immoral to do business on specific dates. In other cases, the ethical requirements might make different demands to the system, but one such set of demands is more feasible than the other; here, the best problem formulation will be driven by the *consequences* of the possible sets of ethical constraints. Finally, it is possible that significant sets of stakeholders hold values that are mutually incompatible in their absolute forms, without any set being clearly preferable to another; any solution, then, will probably raise ethical issues for some stakeholders, and finding a working compromise is only possible by addressing on a way or another the concerns of all parts.

The VSD framework can be useful to find a solution that addresses such valorative concerns. Drawing from Davis [2009], this article proposes value scenarios — narratives that explore how people may use a system such as a simulation, identifying effects of the designed system on direct and indirect stakeholders over varying periods of time — as a tool to identify not only the critical values, but also relevant value clashes, such as *value sinks* strongly opposed by a subset of the relevant actors, and *value flows* that are not critical to the system itself but might be beneficial

due to their widespread acceptance. The results of those scenarios should then be empirically validated through the usual empirical software engineering approaches.

After identifying the relevant ethical issues relevant for simulation design, it is necessary to incorporate them into the logical model of the target phenomenon. That model can be either a closed mathematical system or a set of rules without a closed form, such as the interaction rules of an agent-based model [Elsenbroich and Gilbert 2014]. In this stage, the main concerns are related to model opacity, not only from the mathematical standpoint but also due to institutional constraints [Burrell 2016]; an answer to these forms of opacity is the explanation standard described by Doshi-Velez and Kortz [2017]: it should be possible to predict, given a set of inputs, how a specific factor can affect the output, based on the explanation requirements of stakeholders, identified for example through participatory design techniques [Davis 2009].

A valid logical model must then be implemented into an actual computer system. Since the software development processes involve specific technical expertise, including the relevant stakeholders in this stage might prove difficult, intensifying the opacity resulting form technical illiteracy. As a consequence, developers must take increased care on this stage, taking care to add as little opacity as possible to the logical models — or even introducing means for understanding the inner workings of the system, such as visualizations and result reproducibility —, while making sure that the models can be altered after construction so as to address issues identified during validation and also from the feedback obtained after deployment.

Procedures for ethical data collection are the subject of a wide literature that is not specific to the design of computer simulations, as discussed by Alvarez [2016]. Still, simulations themselves might generate significant amounts of data about their subjects, and that information must be stored in ways that do not expose the simulation targets and users to risk and that enable external validation, correction, and exclusion of information, as defined by laws such as the European GDPR. To address the issue of lack of feedback, that data must also be comparable to external outputs that allow simulation users and designers to identify prediction errors, which might prompt model redesign or a change in use cases.

Verification and validation processes can be a critical stage in ensuring that simulations behave in a proper way. In a technical sense, V&V processes should ensure that the resulting computer system can produce results within tolerable error margins and that it can handle any reasonably expected set of parameter and input values and operate at the required scales [Kaner and Swenson 2008]. Ethical hazards then can be treated as part of the validation criteria, to be evaluated in test cases with methods similar to those used for value extraction, and addressed at each stage of the cycle.

Validation by the team itself can be supplemented by external evaluations of the simulation design and use cycles, which can be conducted by regulatory organs, NGOs, or other trustworthy sources. An external perspective should allow the design team to address their own biases, but business, legal, or other demands might introduce secrecy requirements. This demand might be partially addressed through non-disclosure agreements and similar controls, and the combination of constant internal evaluation and external valuation will allow simulation designers to adjust simulation outputs to

reflect the actual results from the simulation-based interventions and adapt the constructed model to meet the ethical demands of stakeholder interactions.

4. A checklist for ethical simulation design

In this section, we synthesize the actionable insights from the previous section into a checklist for inspecting the development cycle of computer simulations. Checklists are an established tool for software inspection [Brykczynski 1999], which lend themselves easily to the evaluation of the entire simulation design; instead of finding software defects, this checklist will point sources of ethical hazards that should be inspected.

1. Is there an initial mapping of background relations involving the simulation?
2. Is there a clear sampling process for selecting background stakeholders to take into account?
3. Is there an unified model that reconciles the ethical demands of direct and background stakeholders?
4. Do the functional and nonfunctional requirements of the proposed system reflect the ethical demands from background stakeholders?
5. Is it possible to predict how variations in individual variables and parameters will affect the simulation output?
6. Is there a clear model of the possible sources of ignorance about the simulation results?
7. How does the computer implementation of the logical model add new sources of opacity to simulation?
8. Are there any interaction effects introduced by running the simulation with a large dataset and/or for a significant timespan?
9. Can the resulting simulation system be reasonably modified to handle new ethical constraints identified after its construction?
10. Are simulation results produced in a way that can be reproduced and compared with external data?
11. Can external stakeholders, directly or through representatives, propose changes to the finished model so as to address ethical concerns?

The presented checklist draws from good practices on checklist design that require non-trivial verification effort [Brykczynski 1999], but further work is required for empirical validation of the suggested practices. Still, the modular nature of the checklist means that it can easily be extended, shortened or altered based on the practical feasibility of this initial proposal.

5. Concluding remarks and further work

Whenever computer simulations are employed as decision-aiding tools in problems with social consequences, their use happens within ethical, legal and other forms of social constraints, which can either affect the means used by a simulation to achieve its objectives or even prevent that accomplishment at all. Still, the relationship between a social simulation and its running environment must not be seen as one-sided: through direct and indirect interactions with stakeholders, computer simulations can shape the ways in which people perceive their environment and act on it.

In such a context, simulation developers must be prepared to address the valid concerns of stakeholders that at first may seem to be significantly removed from the scenarios in which simulations are used. Respecting the needs and desires of those remote stakeholders may bring additional technical or social burdens to the simulation, such as the need for explaining the effect of each input variable. Yet, failing to do so might not only be unethical — by trampling the rights of other human beings without their consent or sometimes even their knowledge — but also illegal, resulting in blowback to systems designers and users.

This paper adopted a restrained approach, pointing out what kind of ethical demands are placed upon simulation developers by computer ethics and the philosophy of technology, and then showing how those requirements can be addressed within the usual software development processes. Further work is needed, especially for identifying relevant metrics for the explanation of simulation outputs and relevant stakeholder coverage, but the association between ethical concerns and established software engineering practices allow development teams to address the key concerns of system opacity, lack of feedback from real-world results, and lack of understanding of large-scale effects that have already caused unjust results in many spheres of contemporary societies.

Acknowledgements

This work was partially funded by CNPq/Brazil (305621/2015-7). The authors would also like to thank three anonymous referees for their feedback.

References

Alvarez, R. M. (2016). Introduction. In: Alvarez, R. M. (ed.) Computational Social Science. Cambridge University Press, 2016.

ACM US Public Policy Council. (2017). Statement on Algorithmic Transparency and Accountability.

Brykczynski, B. (1999) A survey of software inspection checklists. ACM SIGSOFT Software Engineering Notes, 24(1), 82.

Burrell, J. (2016). How the machine ‘thinks’: Understanding opacity in machine learning algorithms. Big Data & Society, 3(1), 2053951715622512.

Cioffi-Revilla, C. (2014). Introduction to computational social science. London and Heidelberg: Springer.

Davis, J. (2009). Design methods for ethical persuasive computing. In Proceedings of the 4th International Conference on Persuasive Technology. ACM.

De Marchi, S. (2005). Computational and Mathematical Modeling in the Social Sciences. Cambridge University Press.

Di Paolo, E., Noble, J., and Bullock, S. (2000) Simulation Models as Opaque Thought Experiments. In: Seventh International Conference on Artificial Life. Cambridge: MIT Press, 497-506.

Doshi-Velez, F. and Kortz, M. (2017). Accountability of AI Under the Law: The Role of Explanation. Berkman Klein Center Working Group on Explanation and the Law.

Elsenbroich, C. and Gilbert, N. (2014). Modelling Norms. Springer, 2014.

Eubanks, V. (2018) Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor. St. Martin's Press.

Floridi, L., and Sanders, J. W. (2002). Mapping the foundationalist debate in computer ethics. Ethics and information Technology, 4(1), 1-9.

Gilbert, N. and Troitzsch, K. G. (2005). Simulation for the Social Scientist. Open University Press.

Hvistendahl, M. (2018) You are a number. Wired 26(1): 48-59.

Kaner, C. and Swenson, S. J. (2008). Good Enough V&V for Simulations: Some Possibly Helpful Thoughts from the Law & Ethics of Commercial Software. In Proc. Simulation Interoperability Workshop.

Koabel, G. (2017) Simulating the Ages of Man: Periodization in Civilization V and Europa Universalis IV. Loading..., *10*(17).

Mackenzie, D. and Spears, T. (2014) 'The Formula That Killed Wall Street': The Gaussian Copula and Modelling Practices in Investment Banking. Social Studies of Science, v. 44, 393–417.

Ohm, P. and Reid, B. (2016). "Regulating Software When Everything Has Software," 84 Geo. Wash. L. Rev. 1672-1702.

O'Neil, C. (2017). Weapons of math destruction: How big data increases inequality and threatens democracy. Broadway Books.

Ören, T. I., Elzas, M. S., Smit, I., and Birta, L. G. (2002). A code of professional ethics for simulationists. Proc. 2002 Summer Computer Simulation Conf. (San Diego CA).

Pasquale, F. (2015). The Black Box Society: The Secret Algorithms That Control Money and Information. Harvard University Press.

Sen, A. (2011). The Idea of Justice. Belknap Press.

Siegfried, R. (2014). Modeling and Simulation of Complex Systems: A Framework for Efficient Agent-Based Modeling and Simulation. Springer Viehweg.

Sommerville, I. (2011). Software Engineering, 9th ed. Addison-Wesley.

Stahl, B. C., Timmermans, J., and Mittelstadt, B. D. (2016). The ethics of computing: A survey of the computing-oriented literature. ACM Comput. Surv. 48, 4.

Verbeek, P.-P. (2005). What things do; philosophical reflections on technology, agency, and design. Pennsylvania University Press.

Williamson, T. J. (2010). Predicting building performance: the ethics of computer simulation. Building Research & Information, 38(4), 401-410.

Yu, L., Li, X., Tang, L., Zhang, Z., and Kou, G. (2015). Social credit: a comprehensive literature review. Financial Innovation, 1(1), 6.