Standing on the Shoulders of Giants—And Then Looking the Other Way? Epistemic Opacity, Immersion, and Modeling in Hydraulic Engineering

Matthijs Kouw
PBL Netherlands Environmental Assessment Agency
VU University, Institute for Environmental Affairs

Computational models are the predominant instruments for studying water-related phenomena in hydraulic engineering. Over time, these models have become more and more opaque, making it more difficult for modelers to grasp their functioning. As a result, both model developers and users straddle discovery and manipulation, since they may not be able or willing to reflect on how computational models shape their understanding of the world. Hydraulic engineers often engage models in a reflective manner that is aimed at understanding a model's underlying design. However, models in the form of software travel easily to domains outside of hydraulic engineering, where they are not used in a reflective manner. In order to prevent that model users become subject to the will-o'-the-wisp of opaque computational models, the aforementioned reflective approach to modeling warrants adoption by model developers as well as model users.

1. Introduction: Blackboxing and Epistemic Opacity

Over the course of the twentieth century, hydraulic engineering has come to rely primarily on the use of computational models. Disco and van den Ende (2003) hint towards the reasons for widespread adoption of computational models by pointing out that such models fulfill a crucial role as management tools in Dutch water management, and meet a more general desire to quantify water-related phenomena. The successful application of computational models implies blackboxing (Latour 1987 and 1999): "[w]hen a machine runs efficiently … one need focus only on its

Perspectives on Science 2016, vol. 24, no. 2

doi:10.1162/POSC_a_00201

inputs and outputs and not on its internal complexity. Thus, paradoxically, the more science and technology succeed, the more opaque and obscure they become" (Latour 1999, p. 304). The successful application of blackboxed technologies, in this case computational models, means they are taken for granted and usually only come into view when failure or malfunctioning renders them obtrusive.

In hydraulic engineering, blackboxing can imply that simulationists[1] are less likely to reflect on the design of models and how their understanding of the world is shaped accordingly. Turkle (2009) claims that modeling features an increasing degree of immersion, which can be defined as an engrossing, enticing, or captivating influence of technologically mediated practices and experiences (see also Calleja 2011; Causey 2009). Immersed in modeling practices, simulationists straddle discovery and manipulation: "[t]he more powerful our tools become, the harder it is to imagine the world without them" (Turkle 2009, p. 8). Thus, a simulation may "propose itself as proxy for the real" (Turkle 2009, p. 80) Turkle argues that maintaining a "deep connection between hand and design" (2009, p. 16) allows technological mediation to be lived rather than accepted at face value. Immersion implies that simulationists are not concerned with how models shape their findings by functioning as proxies for the real, and accept model output without batting an eye.[2]

Following Turkle's diagnosis of immersion related to simulation, the famous adage "all models are wrong, but some are useful" (Box and Draper 1987, p. 424) should be replaced by "all models are wrong, but some are dangerous." The fact that models function as proxies for the real need not be a problem in and of itself. However, a failure to understand how they function as proxies for the real may have disadvantageous effects. For example, models in hydraulic engineering may be based on false assumptions, equations or input data, which may cause technological malfunctioning, unnecessary improvements to flood defenses, or erroneous foundations of water-related policies.

Blackboxing and immersion are related in that blackboxing renders technologies opaque, which can lead to or exacerbate immersion: it is difficult for simulationists to grasp how opaque models function as proxies for the real. How can opacity be understood? Paul Humphreys attributes

---

[1] Winsberg (2010) and Petersen (2012) use the term simulationist when speaking of scientists, engineers, and other social actors who engage in the development and/or use of simulations and models.

[2] These concerns can be aligned with the "Special Issue on Simulation, Visualization, and Scientific Understanding" of *Perspectives on Science* published in the fall of 2014, which discusses how simulations contribute to scientific understanding.

epistemic opacity to a model in the case where it is "impossible" for a cognitive agent "to know all of the epistemically relevant elements of the process" (Humphreys 2009a, p. 4). Humphreys' notion of epistemic opacity concerns both the instrument and its user since it influences the *ability* and *willingness* of a cognitive agent to "know that what the instruments display accurately represents a real entity" (2009a, p. 4). The underlying complexity of models and their functioning according to expectations can reduce the likelihood that cognitive agents can or will question the models at their disposal. Epistemic opacity can imply immersion insofar as it reduces the likelihood of understanding how a particular model acts as a proxy for the real. Humphreys claims the epistemic opacity of computational models implies that their relationship with the world cannot be fully comprehended by individual cognitive agents.[3] In practice, this means modeling has become a "social epistemology, within which the work has to be divided between groups of scientists or mathematicians, so that no one person understands all of the process" (Humphreys 2009a, p. 5).

Although computational models have become increasingly opaque, this does not necessarily imply immersion. Grüne-Yanoff and Weirich claim that simulations can only be used legitimately in case of "analytic understanding of at least the underlying mathematical equations" and propose an "experimental approach to simulations," which consists of "a strategic move to 'black-box' (Dowling 1999, p. 265) the known program and to interact 'experimentally' with the surface of the simulation" (Grüne-Yanoff and Weirich 2010, p. 26). Some simulationists appear to interact experimentally with the model's surface, which involves changing input parameters, tweaking the model to see how it responds to parameter changes in the underlying calculations, and making sure the model's output can be verified by one's personal expertise. Thus, simulationists attempt to grasp the inner workings of models, making it less likely that their use of models is accompanied by immersion. In the following, I will use the terms 'reflectivity' and 'reflective approach to modeling' to indicate the aforementioned process of 'experimental interaction with the model's surface'.

1.1. Research Questions and Methodology

How and to what extent do reflective approaches to epistemically opaque models prevent immersion? I address these research questions by first

---

[3] Humphreys uses the concept of epistemic opacity to denote emergent features of modeling practice that reveal their inexorable prosthetic nature as knowledge instruments that cannot be fathomed entirely: running models generates emergent macro-level features that would not appear without the use of simulations. This requires new macro-level descriptions that will be able to capture these features (Humphreys 2004, 2009a, 2009b).

describing how models used by simulationists have become increasingly opaque, and how hydraulic engineers deploy reflective approaches as a strategy to cope with epistemic opacity. Subsequently, I illustrate the tendency to abstain from reflective approaches to modeling, which is due to the codification of knowledge in the form of computer software that can travel easily to domains outside of hydraulic engineering. By addressing these issues, I assess the danger of epistemic opacity as a possible cause for immersion and indicate the importance of reflective approaches to modeling. Given the observation that epistemically opaque models can become dangerous due to immersion, and the fact that social reliance (Pippin 1995, p. 46) on modeling is increasing, I believe such questions warrant substantial attention.

This paper draws on a total of 28 semi-structured qualitative interviews conducted between 2009 and 2011. These interviews were held at Deltares (Delft, The Netherlands), a Dutch institute for applied research in the field of water, subsurface and infrastructure. Additional interviews were held at VORtech (Delft, The Netherlands), a software development company that specializes in model development and maintenance, and Radboud University (Nijmegen, The Netherlands), where an interactive model discussed in section 3 was being developed.

This material enables an examination of the relationship between epistemic opacity and immersion from various angles. Hydraulic engineers at Deltares have witnessed profound shifts in the materiality and organization of modeling that have rendered models opaque. In addition, modeling activities at Deltares have led to the development of modeling software where earlier successes are taken as a starting point. As a result, an already large codebase grows more and more opaque over time, posing challenges to parties like Deltares and VORtech. The expertise of hydraulic engineers is codified in the form of computer software, which can travel easily to domains of practice where the aforementioned reflective approach to modeling is not adopted, as is witnessed by my interviewees from VORtech and Radboud University Nijmegen.

2. Reflective Approaches to Modeling in Hydraulic Engineering

Computational models have become more opaque due to technological innovations, which have shaped modeling practices profoundly. However, hydraulic engineers stress the need for reflective approaches to modeling as a necessary response to these changes.

2.1. Computational Prowess

Increases in computational power fueled the adoption of computational models at Deltares, since they allowed phenomena to be modeled on high resolutions, the use of large data sets, and performing multiple runs of a model. Among hydraulic engineers, it is still undecided whether scale models can be fully replaced by computational models. One supposed advantage of scale models is that they bear a physical resemblance to the phenomena they are meant to simulate, provided scaling effects are taken into account.[4] In the literature, this advantage has been subjected to scrutiny (Morgan 2003; Parker 2009; Winsberg 2009), though several hydraulic engineers at the Coastal Structures Department at Deltares attribute importance to physical modeling in terms of accuracy and detail. Water movements near coastal structures and coastlines can be characterized as turbulent due to the interactions between water and structures or land. In such cases, the effects of water on coastal structures or coastlines may be difficult to predict, since the behavior of the water may contain little to no patterns, or that these patterns turn out to be difficult to find. Computational models that fail to describe these interactions can put coastal structures at risk. As a result, physical models can provide important means to validate and calibrate computational models, though it is not self-evident that computer simulations will not reach similar levels of accuracy and detail in the future.

This reluctance to embrace computational prowess as a premonition of more elaborate forms of modeling is due to a desire to incorporate models into one's preferred way of working, which is based on personal expertise and practical concerns rather than dictated by technological possibilities. As the head of the Coastal Structures department stressed: "you just look for the right tools on a case by case basis" (Interview June 4, 2009). The pragmatic tendency of hydraulic engineers to look for the right tool for the job at hand also becomes apparent in their ideas about model resolution. Although increases in computational power enable models of phenomena in higher resolutions, it is not always possible for models to tap into the potential of increased computational power. Models may be developed for a particular purpose and may deliberately represent a target system in a simplified manner. Making sure the model performs equally well on a higher resolution may require a different approach altogether, such as a different set of underlying equations. In addition, models may contain lines of code that cannot simply be omitted or modified. For example, a model may contain parts of FORTRANM[5] code, which are usually left alone without being subject to thorough evaluation due to a lack of time and expertise.

---

[4] For example, the behavior of water tends to change on a smaller scale due to a different influence of gravity on water movement, and water has a different viscosity on smaller scales.

[5] A programming language that was first developed in the 1950s and became prominent in scientific computing in areas such as fluid dynamics and meteorology.

In addition, hydraulic engineers may not make use of the possibilities afforded by increases in computational power due to considerations concerning model performance and accuracy. Jaco Stout of Deltares points out that hydraulic engineers will often try to push the envelope. If increases in computational power allow more detailed calculations and more model runs, hydraulic engineers may attempt to make their models more detailed, provided it does not take too long for a model to produce its output.[6] In hydraulic engineering, long calculation times may not be acceptable due to limits related to time and resources, but also because hydraulic engineers want to have the ability to do multiple model runs. Input values typically have a great effect on the output of a model. Therefore, studying the effects of different input values is of great importance. Hydraulic engineers may wish to calibrate models by using different input values, and make adjustments in the model's code or schematization where necessary.

2.2. Keeping Models under Control

Gerben de Boer, Senior Researcher at Deltares, explains that the ability to do multiple model runs is a matter of keeping the model "under control."[7] De Boer mentions three rules of thumb for keeping calculation times in check: model runs should take either five minutes (the time it takes to get a cup of coffee), one night (so you can check the results in the morning), or two days to a week (so you can run the model on Friday and check the results after the weekend or, say, a skiing trip). De Boer refers to calculations that take longer than a week as

> "count your blessings" calculations … it is interesting to see what comes out of it, but you cannot rely on it … a model can perform calculations, but you have to understand what is happening … you have to know what you are looking for, so you have to understand the underlying physics. The model cannot replace your insights that are based on physics. (Gerben de Boer, interview by author, June 19, 2009)

For de Boer, keeping a model under control implies knowing and understanding the equations underlying the model in question. The model's design should be brought to bear on one's expertise as an engineer.

Without a doubt, running multiple and complex model calculations has tremendously impacted the work of hydraulic engineers. During the early days of computational modeling, one often had to file a request for a

---

[6] Jaco Stout, interview by author, June 19, 2009.

[7] Gerben de Boer, interview by author, June 19, 2009.

particular calculation at a designated department and then wait for the results. The slightest mistake in the schematization, code, or mathematical calculations underlying a given model would result in model output that could not be used. Adri Verwey, Senior Specialist Modeling Systems at Deltares, who has worked on a variety of hydrological and hydrodynamic models since the 1970s, points out how increases in computational power could impact one's ability to keep a model under control. During his student years, Verwey was doing research on the only mainframe available at the university where he was working at the time. Verwey admits the required concentration and attention to detail could be a nuisance. However, he also points out this made him think very carefully about the way the model was set up, and made him pay attention to the limitations of the model. When the opportunity arose for him to use the mainframe for an extended amount of time on a quiet Sunday, Verwey noticed that he was "just doing some calculations" and was not really paying the attention normally required to ensure the quality of the model's output.[8] Here, Verwey distinguishes 'just doing some calculations' from 'paying attention to ensure quality of output' in order to stress the need of ensuring that one has a grasp of a model's structure, rather than playing around to see what happens. As was the case with de Boer, bringing one's own expertise and understanding to bear on a model's underlying structure is seen as a way to keep the model under control and ensure that its output is valuable. More generally, hydraulic engineers stress the importance of aligning models with their expertise. When they do not understand the target system, base the model on ill-founded assumptions, or fail to comprehend the intricacies of a given model's design, the hydraulic engineers at Deltares fear they end up prematurely using models that they do not really understand.

2.3. Models as Sparring Partners

Many hydraulic engineers at Deltares stress models cannot prove anything, and that they can only show the consequences of your own assumptions. In other words, models do not provide an easy way to understand how things work, but can only confirm whether things work in the way you thought they did beforehand. Any model is first and foremost a simplification of reality developed with a certain purpose in mind. As a simplification of a more complex system, a model may generate useful insights. Models are in this sense question-driven, meaning that hydraulic engineers need

---

[8] Adri Verwey, interview by author, May 27, 2009.

to know what they want to find out when constructing a model. As Edward Melger, Product Manager at Deltares, explains:

> You only construct a model once you have a question you want to answer … if the question is not clear, you can have a very nice model that delivers a beautiful answer, but it can never be correct. The question first needs to be clear. (Interview Edward Melger, May 26, 2009)

Similarly, Gerben de Boer argues that the output of a numerical model is merely an advanced version of a back of the envelope version of a target system, namely a rough schematization and characterization of a target system based on expertise. The model can only provide more detail, not a radically different answer or a profoundly deeper understanding of phenomena: "if you do not know, roughly, what comes out of it beforehand, you do not need to run a model. I would say a numerical model is nothing more than a refinement of something you can do yourself."[9]

In case of more exploratory use or a mismatch between a modeler's own ideas and model output, a numerical model functions more like what de Boer refers to as a "sparring partner" in the sense that a model might challenge your own thinking and lead to new insights:

> You have to know what your question is beforehand, so if you do not know what your question is, the model is more like a sparring partner that might be able to tell you something about the system." (Gerben de Boer, interview by author, June 19, 2009)

In other words, not having a clear idea of one's question is not necessarily a reason to put a model aside, since it can function as a sparring partner, making the model a conduit to new insights. However, this is a slippery slope that could lead to immersion; both Melger and de Boer stress the need to understand and be aware of the question underlying a model's design. As de Boer points out, a model "might tell you the wrong things about the system because it has been constructed for a different purpose."[10]

Sometimes a modeler needs to take a step back to study whether a model can indeed be used to answer a particular question. For example, a two-dimensional model of a lake can be used to model water levels, but not for modeling sediment transport since such events involve three-dimensional processes where different layers of water interact in turbulent processes. Successful applications of a model in one problem area by

---

[9] Gerben de Boer, interview by author, June 19, 2009.

[10] Gerben de Boer, interview by author, June 19, 2009.

no means guarantee similar successes in other problem areas. Simulationists should therefore gain familiarity with the model they are using in order to understand the implications of its use, for example, by studying the questions that informed the construction of a particular model, how the model is built, how the model's schematization relates to its target system, and what data is used as input for the model in question. Grasping the model's design through reflective use is seen as a necessary aspect of assessing whether a model can be used to address a particular question.

Understanding these aspects of a model enables the modeler to have a degree of control over his or her instruments, which the hydraulic engineers quoted above see as a necessary precondition to credible use of models. Verwey argues hydraulic engineers should develop their expertise by reflecting on the models they are using. He admits his own career puts him in a rather fortunate position in this respect, since he experienced the very early stages of model development, which provided him with knowledge of design principles on which many subsequent developments are based.[11] The younger generation of hydraulic engineers often simply does not have the ability to study the design and deeper foundations of models in such detail, and is, in that sense, often condemned to using models out of the box. Some modelers I encountered at Deltares have spent years, even decades, working on one particular model. Although different generations of modelers may have different degrees of familiarity with a model's design, it is certainly not the case that younger generations of modelers are no longer interested in understanding the underlying design of their models. In fact, making sure one has some knowledge of a model's basic structure is considered good modeling practice.[12]

2.4. Social Epistemologies

Modeling increasingly involves groups of people rather than individuals, which points to Humphrey's notion of social epistemologies discussed in the introduction. Software engineers often carry out the development and maintenance of models initially designed by hydraulic engineers, distributing modeling practice over an even larger and more varied group of social actors. Over time, modeling has also become firmly intertwined with policy making. The perceived reliability of models due to successful applications in the past has also increased the complexity of challenges they are

---

[11] Adri Verwey, interview by author, May 27, 2009.

[12] Turkle (2009) elaborates on the uncritical adoption of simulations by younger generations. However, who these users are, why they act in the various ways they do, and what simulations and models they have at their disposal remains unclear. As a result, Turkle's discussion of present-day generations of simulationists displays an unnecessary amount of pessimism, perhaps even resentment.

expected to address. As a result, more intricate and elaborate models of larger systems need to be developed.

Hydraulic engineers at Deltares claim the opacity of models appears to be increasing due to the dispersion of modeling and demand for models addressing complex issues, and stress the importance of understanding how model outputs is produced. Rather than trusting a model out of the box, they argue, one should become familiar with a model in an exploratory manner, e.g., by starting with relatively simple phenomena, such as the discharge of a large river. Other aspects of the river can then be added incrementally, leading to the study of more and more complex phenomena. Accepting the model's design and using it immediately to address highly detailed and complex issues is unacceptable and tantamount to irresponsible use of models. According to this line of reasoning, modeling practice requires a reflective approach in which hydraulic engineers dedicate themselves to becoming familiar with a given model. However, the dispersion of modeling practice over a larger and more varied group of social actors and the fact that modeling practice faces increasingly complex challenges do not bode well for reflective approaches to modeling as attempts to grasp the design of models and the implications of their use.

3. Traveling Knowledge

Knowledge produced by hydraulic engineers is codified in the form of computer software that can travel easily to domains outside of hydraulic engineering. This entails a tendency to abstain from reflective approaches to modeling, which may very well enhance the risk of immersion as a result of epistemic opacity.

3.1. Progressing Understanding and the Codification of Knowledge

Increases in computational power are integrated gradually and reflectively on the basis of the expertise of hydraulic engineers and the specificities of the task at hand. In some cases, the use of highly simplified representations of target systems can be justified when they capture all of the physical processes that are considered to be relevant. Hydraulic engineers use computational models in terms of *sufficiency*, not *accuracy*. As Karel Heynert, Head of the Hydrodynamics and Operational Systems Group at Deltares, points out somewhat ironically, his work is about finding solutions for problems, and not vice versa.[13] The importance attributed to being familiar with a model's design shows how hydraulic engineers at Deltares attempt to find the right tool for the job at hand, while making sure they use that tool responsibly at the same time.

---

[13] Karel Heynert, interview by author, June 10, 2009.

Hydraulic engineers at Deltares still abstain from stipulating future successes of models. For example, they point out that it is difficult to make hard claims about the progress or reliability of models, since the issues they are meant to address and their application areas co-evolve with societal demands. In a similar vein, Edward Melger admitted that models can be applied successfully in the study of certain issues, but understands their value in terms of the insights models can provide to hydraulic engineers. Melger also does not think models are approximating reality more and more since they are never completely exhaustive. Rather, modeling consists of balancing the questions models are supposed to address against the model's possibilities and measurement data that happen to be available.[14]

In this perspective, modeling practice has a provisional character in which the ability of models to capture fundamental principles or law-like structures of reality is not an objective. According to Simone van Schijndel, Manager of Operational Water Management group at Deltares:

> it is not so much the case we are not interested in that, but rather that we realize it is not possible, and that is where I think there is a discrepancy with the policymaker, who I think does consider it to be possible … we are well aware of the fact that is just not reality, which is out there, not here in the computer (Interview Simone van Schijndel, June 24, 2009).

Since most policy makers demand clear-cut answers, van Schijndel had to withstand a lot of critique when she wrote reports that stressed the need for more research in order to deal with uncertainties. For her, modeling is much more about making abstractions with a particular purpose in mind, and often not about making more and more accurate approximations of reality. Even measuring what happens 'out there' in reality is problematic since the behavior of real systems often contains noise, such as passing ships or storms. "So you have to construct a model. At the same time it is very crucial that your model describes accurately what happens … so in that sense you arrive in a paradox, or a deadlock, what is reality?"[15] Following van Schijndel's words, reality is elusive, and one cannot expect models to fully describe reality. However, this is not necessarily a problem in itself, as long as one realizes the purpose underlying the model in question.

To sum up, hydraulic engineers at Deltares may speak of the reliability of models in terms of progressing understanding. However, this claim is based on practical results, and does not necessarily diminish their reflective approach to modeling. That said, hydraulic engineers do appear to believe

---

[14] Edward Melger, interview by author, May 26, 2009.

[15] Simone van Schijndel, interview by author, June 24, 2009.

computational models are becoming more successful in terms of understanding and predicting hydrological and hydrodynamic phenomena. The hydraulic engineers at Deltares often refer to this as progressing insight, which might appear to contradict their reflective attitude towards models. The reliability of models is not so much explained in terms of their ability to yield an objective understanding of the world, but rather in terms of heuristic currency that is based on practical results. The hydraulic engineers at Deltares continue to adopt a reflective attitude towards computational models, but also value strong correlations between model output and measurements as proof they can trust the model in question. The history of Deltares and its many successes, which were to a major extent based on models, are frequently mentioned as a source of this trust.

3.2. Integrated Water Management and Modeling Interfaces

Policymakers have come to demand models that address various aspects of complex systems (e.g., interactions between spatial planning, flood protection, and ecology). In addition, the perceived success of models, which is based on successful application in the past, has led to the codification of expert knowledge: once models have proven their value in the past, they are used as building blocks for subsequent model development. Both the desire to model large and complex systems and the perceived success of models agave rise to integrated water management, which is a broader tendency of policymakers to address social, economic, and ecological aspects of water management simultaneously (Martinez-Santos et al. 2014).

A concrete example of codified knowledge can be found in so-called modeling interfaces, which enable the construction of modeling infrastructures in which modular model components can be exchanged between various parties. Modeling interfaces are elaborate protocols that ensure that model components meet certain requirements, allowing them to be connected to each other in more elaborate models. The model components in question are based on knowledge derived from models that were successfully applied in the past, and have withstood reviewing procedures. As software, these model components allow model-related expertise to travel easily from one domain of practice to another.

The development of modeling interfaces was considered necessary to facilitate integrated water management, which requires the modeling of individual bodies of water as well as their interactions with other systems. The development of OpenMI (Open Modeling Interface) is supervised by the OpenMI Association, which took as its starting point the observation that the construction of a single all-encompassing model of all relevant bodies of water would be too costly. In addition, such a model would require a laborious process of negotiation between various parties involved, large amounts of computational resources, and would lead to a model that would be difficult to maintain and understand, exacerbating epistemic opacity. Finally, the OpenMI Association also wishes to enable more flexible forms of simulation practice.

According to Gregersen et al., integrated water management "requires the linkage of individual models or model components that address specific domains … the OpenMI has been developed with the purpose of being the glue that can link together model components from various origins" (2007, p. 175). By acting as a glue between model components, OpenMI provides adaptability of model components that enables the migration of existing modeling systems, which is important "since their re-implementation may not be economically feasible due to the large investments that have been put into the development and testing of these systems" (Gergerson et al. 2007, p. 175).

OpenMI enables integrated water management by providing a protocol that enables interactions between different model components. Simulationists can develop integrated models by connecting model components, provided these meet the requirements of the OpenMI protocol, which thereby functions as an interface. These model components can then exchange data during run-time. An everyday example of an interface would be the USB interface (commonly recognized by the small horizontal plugs on the end of cables of mice, keyboards, etc.), which allows users to connect a variety of devices to their computers, provided these devices meet the necessary requirements. OpenMI-compliant model components that end up in an integrated model can be developed by different parties, represent different processes related to different problem areas (e.g., hydrology, hydrodynamics, ecology, economics, etc.), and may use different dimensionalities (e.g., 1D, 2D, 3D models), modeling principles (e.g., deterministic, stochastic, static), data sources, spatial and temporal resolutions.

The interconnectivity between the components of an integrated model is facilitated by the OpenMI interface and guarantees adaptability. Model components complying with the OpenMI standard can exchange data during run-time, enabling the creation of integrated models using components from different providers that are considered best suited to the task at hand. Thus, models can be linked "with the minimum of re-engineering and without requiring unreasonably high level IT skills" (Moore et al. 2010, p. 11). The requirements of OpenMI are enabling in the sense that they enhance flexibility and interactivity between model components. However, OpenMI (and interfaces more generally) are also constraining, since they are standards that "impose and enhance particular workflows, thought modes, and modes of interaction upon or in combination with human users" (Cramer and Fuller 2008, p. 151). Documentation on OpenMI stresses it is an open standard, since its specification and source code are freely available on the Internet and that it enables connections between different kinds of models, disciplines, and domains.[16] Thus, simulationists using OpenMI compliant models "will be able to 'mix and match' models from different sources" (Moore et al. 2010, p. 8). Model components can be integrated on an ad hoc basis, without formal cooperation among modelers. As a result, engineers may no longer be able to or have the desire to critically reflect on the design of integrated models built using OpenMI-compliant model components.

3.3. 'Code Drift'

During a presentation I gave at VORtech, several software engineers referred to the aforementioned mix and match approach enabled by OpenMI as shopping. OpenMI enables the exploration and exchange of model components on epistemological bazaars. The software developers at VORtech indicated that OpenMI allows a pragmatic approach in which it is not always possible, nor considered necessary to fully fathom the design of all components of an integrated model. In principle, OpenMI enables the components of an integrated model to exchange data, but in practice it is important to think carefully about the compatibility of model components. Formally, model components are able to exchange data when they are OpenMI-compliant, but the engineers at VORtech do not consider this a guarantee for good results. Some of these model components may be based on radically different approaches to modeling, which makes it crucial to think carefully about the assumptions and ideas that went into them, and the repercussions of connecting these different models. One of the software engineers expressed his concerns about working with OpenMI by pointing out "you are no longer forced to think about the quality of the work of other modelers."[17]

However, other software engineers at VORtech point out that OpenMI establishes code testing as standard procedure, and that adhering to the OpenMI protocol makes one's work accessible to others, who can provide feedback that can be used to make improvements. In this regard, the OpenMI documentation makes an appeal to the responsibility of simulationists: "the OpenMI cannot guarantee that the representation of the process in the component or the link to another component is scientifically

---

[16] Note that open source is different from open standards: open source entails making accessible (parts of) computer code, while open standards apply only to interfaces and agreements related to the exchange of software and/or data. Thus, using open standards may still imply the use of closed software.

[17] Communication with author, June 19, 2009.

valid. That is the responsibility of the modeler, model integrator and user" (Moore et al. 2010, p. 16). Developers of integrated models need to describe what different model components they have linked using a metadata structure that is part of OpenMI, making their design accessible to others. Thus, documentation may counter epistemic opacity, but day-to-day realities of software development often show there is neither time nor a persistent commitment to carefully document code in high detail and with great consistency. In addition, software engineers tend to review code they deem interesting or worthy of attention, despite the fact that code testing is considered good practice.

According to Mark Roest, the Managing Director of VORtech, OpenMI allows the creation of patchworks of models components, which may lead to a fragmentation of their expertise. A developer of an integrated model may know very little about, for example, algae blooms, but may still be able to construct a model that describes such phenomena when he or she uses OpenMI-compliant model components. As a result, both developers and users of integrated models may be less inclined to study phenomena outside of their own domain of expertise. It may also be tempting to use an already existing model component rather than developing one from scratch. However, the range of issues where a model component can be used successfully may be limited. This means that in some cases it might be worthwhile to compare the output of different model components: rather than relying on one single model component, it may be worthwhile to experiment with a variety of model components and compare their output from time to time. An integrated model will generate an answer, but whether that answer is correct can be difficult to find out.[18] An open modeling interface is therefore by no means a guarantee for a reflective approach to modeling.

Integrated models may introduce another risk. The design of models consists of a multitude of different interacting processes, such as formalization, parameterization, discretization, and collecting, parsing, mining, and visualizing data. Choices made at a particular stage of designing a model have repercussions for subsequent stages. The patchwork-like character of integrated models makes it increasingly difficult to fathom their design, and implies the possibility that errors only become apparent when the model malfunctions. As a result, epistemically opaque (integrated) simulations and models may lead to what Snook (2000) has called practical drift. In addition, integrated models echo the concerns advanced by Perrow (1999), who argues the tight coupling and interactional complexity of present-day technologies implies accidents are bound to happen at some point.

---

[18] Mark Roest, interview by author, March 5, 2009.

Paraphrasing Snook, the development of integrated models could imply code drift.

The openness attributed to modeling interfaces deploys a rhetoric that stresses the promising aspects of open source software development, i.e., the exchange of knowledge and expertise and thereby jointly contributing to a collective effort, and escaping the constrictions of commercially developed and proprietary software. Despite the importance of the latter, the patchwork-like character of models built using OpenMI is accompanied by epistemic opacity, which could very well lead to immersion. Integrated models cover up different modeling techniques and may be perceived as properly functioning knowledge instruments.

3.4. Governance Simulations

Water governance involves a variety of issues, such as safety, sustainability, logistics, economics, and the preservation of landscapes with historical value. Water governance is no longer simply a matter of increasing safety: rather than focusing exclusively on preventive approaches to flooding (building, improving, and maintaining flood defenses), approaches to risks have been pushed more and more towards the distribution of responsibilities for harmful events. Present-day political commitments to the development of inclusive water governance and participation entail the desire to extend the use of simulations and models to non-specialists, such as local decision-makers and stakeholders. The reliability attributed to models has led to their application in the realm of water governance.

An example of such governance simulations is the Maptable, a GIS (Geographic Information System) application that allows users to explore the repercussions of water-related policies for various areas in the Netherlands. As the name of the application implies, the model runs on a computer that is embedded in a table. Model output is presented on a touchscreen that occupies a substantial part of the table's surface. The touchscreen can be controlled by means of a keyboard and pen. Toine Smits and Emiel Kater of Radboud University in Nijmegen, who contributed to the development of the Maptable and implemented it in the field, explain that the choice for a table is no coincidence. The table provides a familiar setting that allows different users to stand around the Maptable and negotiate on the basis of the visual output presented on the computer screen. Sitting around a table for the purpose of negotiation and collaboration is thus enhanced. The extended range of water-related issues requires the balancing of more and less compatible problems, such as safety concerns and the preservation of landscape. Kater's views resonate with this more inclusive and all-encompassing form of water governance. According to him, it will ultimately become possible to use the Maptable to study the interactions between hydrological, hydrodynamic, ecological, and economic phenomena.[19] The Maptable allows users to explore and discuss various scenarios related to water governance. The outcome of these interactions can subsequently provide feedback to local decision makers and national policy makers.

Users standing around the Maptable can manipulate the landscape on the Maptable's touchscreen by removing levees, inserting patches of forest, etc. When they have developed a new landscape according to their own ideas, the Maptable calculates the consequences of the proposed changes in the landscape. Within minutes, users can see a visual representation of the consequences of the decisions they have proposed, which also includes dynamic representations in the form of animations. Integrative water governance harbors many different and complex issues, which requires a lot of computational resources and more powerful computers. The developers of the Maptable stress the importance of quickly delivering feedback to users: if it takes too long for Maptable to produce output, users will simply lose interest. Due to the complexity of water governance and the challenge of capturing and keeping the attention of the audience, it may not be feasible to perform highly detailed calculations on the spot.

Though the amount of time it takes for a relatively complex hydrological or hydrodynamic model is small compared to the early days of computational modeling, the need for the Maptable to quickly deliver output requires the simplification of the calculations underlying its representations. Toine Smits and Emiel Kater admit that this might introduce blind spots, but also stress that the main aim of the application is to provoke debate, and certainly not providing elaborate representations. As a result, using the Maptable is not only interactive but also immersive due to the introduction of serious simplifications, which have an impact on the content of participatory water governance that the Maptable aims to establish. A further restriction of the content of water governance is the design of applications running on the Maptable. The various ways in which users can develop scenarios are shaped by decisions made by the Maptable's developers. These decisions might also incorporate the ideas of decision makers and policy makers about water governance into the Maptable's design.

This shaping of integrated water governance leads to the question whether the Maptable's users have the ability to critically engage its design, or express the desire to do so. Although hydraulic engineers may have a more humble expectation of the potential of simulations and models to explore target systems, it is not certain whether users of simulations and models share their point of view. The difference in the priorities of hydraulic

---

[19] Emiel Kater, interview by author, March 25, 2009.

engineers engaged in basic research and those of users working with a particular model may turn out to be difficult to bridge. This may be due to differences in expertise, but also because users of simulations and models may work in a context where a critical and reflexive approach to simulation practice is not always considered important, or may simply be incompatible with the interests of those involved. Uncritical adoption and use of epistemically opaque governance simulations may imply immersion.

Governance simulations are relatively accessible, especially in comparison with earlier forms of modeling that were restricted to hydraulic engineers. However, the accessibility of governance simulations does not necessarily endow an extended audience with a detailed understanding of the various challenges of integrated water governance. This is not a property of software design per se. As Wardrip-Fruin (2009) shows, computer games that remain sufficiently transparent may allow users to gain knowledge of the design of these games and reflect on it. Computer games designed according to this principle "create a surface-level experience that will make it possible for audiences to build up an appropriate model of the system internals" (Wardrip-Fruin 2009, p. 300) This so-called SimCity Effect (named after a popular computer game that is representative of the kind of interaction Wardrip-Fruin discusses here) "leads to audience understanding of the operations of an underlying system" (2009, p. 420). However, I hasten to add that new and improved designs do not necessarily provide a solution for the potentially dangerous effects of epistemic opacity: new and improved designs by no means guarantee different user behavior.

4. Conclusion: Modeling in an Age of Codification

Modeling will inevitably involve some form of inscription as a result of abstraction and/or idealization. It is by virtue of distorting reality in some manner that models allow simulationists to study phenomena otherwise not accessible. If those phenomena were observable, there would be a less immediate need to construct models. Gilbert and Troitzsch identify this matter as a problem of weighing complexity and simplicity: "[t]he best map of the world is the world itself, but unfortunately such verisimilitude will tell us nothing about how the world works" (Gilbert and Troitzsch 2005, p. 20). Models will always imply an approximation of actual systems or hypothetical states of those systems, are developed for a particular purpose, and will achieve credibility if they serve the work of simulationists in a manner deemed satisfactory. Thus, models do not have a straightforward relationship to truth and their target systems that can be captured by identifying their relationship with a presupposed and accessible real world.

The repercussions of these aspects of modeling need to be studied in situ and without jumping to accusations about immersion. Hydraulic engineers at Deltares deliberately simplify target systems in order to produce practical solutions for concrete problems, and struggle to make sense of the models they use. However, they also engage models in a reflective manner in which models are used as a way to generate insights whilst model use is usually coupled to one's personal expertise and understanding. Nonetheless, the perceived reliability of computational models has increased across the board, not only in the eyes of policy makers, but also for hydraulic engineers. The establishment of computational modeling as the method of choice in hydraulic engineering has led to an increase in codification: the act of systematization whereby knowledge is accumulated and organized into a system, for example modeling software that is maintained, distributed, and supported by Deltares. Codification replaces tacit knowledge by explicit articulations of knowledge, and subsumes the work of a highly skilled work force by automated processes that have greater efficiency, can be run at a lower cost. Moreover, codification leads to the blackboxing of models in the form of software, which can travel outside of its context of development to contexts of use where simulationists and other social actors may not be committed to reflective uses of models.

Such social actors feature a lesser degree of inclusion, meaning they work outside of the technological frame (Bijker 1987, 1995) in which models are developed. Technological frames are composed of "the concepts and techniques deployed by a community in its problem solving" and is made up of "a combination of current theories, tacit knowledge, engineering practice (such as design methods and criteria), specialized testing procedures, goals, and handling and using practice" (Bijker 1987, p. 168). The notion of technological frame applies to the interactions between various social actors, who may have divergent opinions about the meaning of a particular technological artifact. Technological frames "can be used to explain how the social environment structures an artifact's design" and "how existing technology structures the social environment" (Bijker 1987, p. 173). Technological frames do not structure the interactions between members of particular social groups completely, since the latter have different degrees of inclusion in technological frames and may be members of more than one technological frame.

Social actors outside of the technological frame populated by the hydraulic engineers at Deltares indeed appear to have different priorities and interests. As I have shown, hydraulic engineers at Deltares persistently try to stay in control of their models. Reflectivity is not an antidote against epistemic opacity, but does imply a form of engagement with epistemically opaque models that can reveal the shortcomings of models. In the absence of reflectivity, epistemic opacity is more likely to imply immersion. As a result of codification, modeling practices have become distributed over a larger and more varied group of social actors who do not always have the desire and/or the ability to question models. As hydraulic engineers at Deltares have lamented, the use of blackboxed software sometimes proceeds in a less reflective fashion.

Both Turkle (2009) and Sennett (2008) lament the disconnection between mind and hand brought about by the widespread technological augmentation of human activities, and stress the need for mastery. However, stressing mastery can be problematic in the realm of modeling for two reasons. First, mastery is not the only possible answer to epistemic opacity. Although hydraulic engineers do not object to codification per se, they do stress it should not lead to naïve acceptance of model output. This perhaps suggests a different explanation of mastery, which stresses the value of engagement and experimental interaction with epistemically opaque technologies. Reflective approaches to modeling warrant more attention since it may be impossible to reverse the trends that have established epistemically opaque models. Although the danger of epistemic opacity was signaled in the early days of software development (see for example Dijkstra 1987), present-day challenges of hydraulic engineering do not bode well for Dijkstra's suggestion to "confine ourselves to the design and implementation of intellectually manageable programs" (1987, p. 26). Today, modeling is a highly distributed collective epistemology deeply intertwined with code and knowledge infrastructures, which does not bode well for individual mastery.

However, it is questionable whether experimental interaction with the model's surface will actually counter epistemic opacity and help to prevent immersion. Reflectivity can only counter the effects of epistemic opacity to an extent, since they take place within the bounds of technological designs that are shaped by institutional and socio-political factors. Reflective practice is bound, meaning that there is always a degree of technoscientific ignorance that accompanies simulation practice. Still, immersion should not be answered by positing the need to master technologies. Reflectivity points to a 'situated making-do' that attempts to counter epistemic opacity. Although the perceived reliability of models can make it less likely their design and functioning will be questioned, an appreciation and cultivation of reflective approaches to modeling may be able to counter blind acceptance.

References

Bijker, Wiebe E. 1987. "The Social Construction of Bakelite: Toward a Theory of Invention." Pp. 159–187 in *The Social Construction of Technological*

Systems: New Directions in the Sociology and History of Technology. Edited by Trevor Pinch and Wiebe E. Bijker. Cambridge, MA: MIT Press.

Bijker, Wiebe E. 1995. *Of Bicycles, Bakelites, and Bulbs*. Cambridge, MA: MIT Press.

Box, George, and Norman Draper. 1987. *Empirical Model-Building and Response Surfaces*. New York: Wiley.

Calleja, Gordon. 2011. *In-Game: From Immersion to Incorporation*. Cambridge, MA: MIT Press.

Causey, Matthew. 2009. *Theatre and Performance in Digital Culture: From Simulation to Embeddedness*. London: Routledge.

Cramer, Florian, and Matthew Fuller. 2008. "Interfaces." Pp. 149–152 in *Software Studies: A Lexicon*. Edited by Matthew Fuller. Cambridge, MA: MIT Press.

Dijkstra, Edsger W. 1987. "The Humble Programmer." Pp. 17–32 in *ACM Turing Award Lectures: The First Twenty Years: 1966 to 1985*. Edited by R. Ashenhurst. New York: ACM Press.

Disco, Cornelis, and Jan van den Ende. 2003. "'Strong, Invincible Arguments'? Tidal Models as Management Instruments in Twentieth-Century Dutch Coastal Engineering." *Technology and Culture* 44 (3): 502–535.

Dowling, Deborah. 1999. "Experimenting on Theories." *Science in Context* 12: 261–273.

Gilbert, G. Nigel, and Klaus G. Troitzsch. 2005. *Simulation for the Social Scientist*. 2nd edn. Maidenhead; New York: Open University Press.

Gergerson, J. B., P. J. A. Gijsbers, and S. J. P. Westen. 2007. "Openmi: Open Modelling Interface." *Journal of Hydroinformatics* 9 (3): 175–191.

Grüne-Yanoff, Till, and Paul Weirich. 2010. "The Philosophy and Epistemology of Simulation: A Review." *Simulation & Gaming* 41 (1): 20–50.

Humphreys, Paul. 2004. *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*. Cambridge: New York: Oxford University Press.

Humphreys, Paul. 2009a. "The Philosophical Novelty of Computer Simulation Methods." *Synthese* 169: 615–626.

Humphreys, Paul. 2009b. "Thinking outside the Brain." *Collapse* 5: 549–569.

Latour, Bruno. 1987. *Science in Action: How to Follow Scientists and Engineers through Society*. Cambridge, MA: Harvard University Press.

Latour, Bruno. 1999. *Pandora's Hope: Essays on the Reality of Science Studies*. Cambridge, MA: Harvard University Press.

Martinez-Santos, Pedro, Maite M. Aldaya, and M. Ramón Llamas, eds. 2014. *Integrated Water Resources Management in the 21st Century: Revisiting the Paradigm*. Boca Raton, FL: CRC Press.

Moore, Roger, Peter Gijsbers, David Fortune, Jan Gregersen, Michiel Blind, Jesper Groos, and Standa Vanecek. 2010. "Openmi Document Series: Part a - Scope for the Openmi (Version 2.0).” https://publicwiki.deltares.nl/…/**OpenMI**+**Scope**+v2.pdf?**version**=1… (accessed 18 April, 2015)

Morgan, M. S. 2003. “Experiments without Material Intervention: Model Experiments, Virtual Experiments, and Virtually Experiments.” Pp. 216–235 in *The Philosophy of Scientific Experimentation*. Edited by Hans Radder. Pittsburgh: Pittsburgh University Press.

Parker, Wendy S. 2009. “Does Matter Really Matter? Computer Simulations, Experiments, and Materiality.” *Synthese* 169: 483–496.

Perrow, Charles. 1999. *Normal Accidents: Living with High-Risk Technologies*. Princeton, NJ: Princeton University Press, 1999.

Petersen, Arthur C. [2006] 2012. *Simulating Nature: A Philosophical Study of Computer-Simulation Uncertainties and Their Role in Climate Science and Policy Advice*. Boca Raton, FL: CRC Press.

Pippin, Robert B. 1995. “On the Notion of Technology as Ideology.” Pp. 43–61 in *Technology & the Politics of Knowledge*. Edited by Andrew Feenberg and Alistair Hannay. Bloomington, IN: Indiana University Press.

Sennett, Richard. 2008. *The Craftsman*. New Haven, CT: Yale University Press.

Snook, Scott A. 2000. *Friendly Fire: The Accidental Shootdown of U.S. Black Hawks over Northern Iraq*. Princeton, NJ: Princeton University Press.

Turkle, Sherry. 2009. *Simulation and Its Discontents*. Cambridge, MA: MIT Press.

Wardrip-Fruin, Noah. 2009. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. Cambridge, MA: MIT Press.

Winsberg, Eric. 2009. “A Tale of Two Methods.” *Synthese* 169 (3): 575–592.

Winsberg, Eric. 2010. *Science in the Age of Computer Simulation*. Chicago: Chicago University Press.