# Whither Visualization Logic

## Toward Data-Driven Behavior

**Leigh Orf[1], Lucas C. Villa Real[2], Gerd Heber[3]**

In many business applications, there is a noticeable tension between greater separation and deeper integration of business data and logic. Using that metaphor, we observe that there is a similar tension in scientific visualization between data to be visualized and visualization workflows, which are representations of the "visualization logic." Trygve Reenskaug, who invented the model-view-controller (MVC) pattern, was aware of this problem, and endorsed subsequent proposals that tried to address it, such as that of "naked objects" or "naked functions" [1]. For several reasons, state-of-the-art visualization frameworks have not adopted these ideas, and we've ended up with a hodge-podge of entangled controllers and views, and plenty of glue code. The ad hoc nature of how applications resolve this tension has profound implications for reproducibility, performance, portability, accessibility, and others. Also, looking at the topics of interest for this workshop, where we "expect a lot more from visualization," for it to be data-driven, it might be time to rethink the logic of visualization and how to overcome the tendency of separating data and behavior in an actual system. We believe that this "rethink" cuts across all topics of interest for this workshop, and we've selected a few examples to illustrate more specific research questions.

Topic 2.a. deals with uncertainty visualization. Uncertainty comes in many forms, including ensembles of outcomes, what-if analyses, noisy data, missing data, doubtful provenance, etc. Visualization tools are not statistics packages, but many statistical packages contain powerful visualization modules. Perhaps glue code can be written to borrow just enough functionality from external packages to get a use case done, but this creates an immediate problem: what domain objects are involved in the presentation and where can that be discovered? Surely, we are not advocating the wholesale incorporation of statistics packages into visualization tools, but a substitute for such glue code that would represent aspects of uncertainty as discoverable domain object behavior

Topic 3.a. deals with visualization for data and technology at the edge. If the complexity of visualizing application domain objects wasn't enough, remote experiments, IoT, sensor networks, etc., appear to introduce other, *non-science application domain objects* that must be accessible to modeling (e.g., uncertainty quantification), analysis, and visualization. It appears that the traditional separation of engineering and science data objects can no longer be maintained, because the correct interpretation of science data depends on the correct interpretation of engineering data (about the experimental setup, etc.). For example, in a scenario involving sensor networks and edge computing, there will be malfunctioning sensors,

[1] University of Wisconsin, leigh.orf@ssec.wisc.edu
[2] IBM Research, lucasvr@br.ibm.com
[3] The HDF Group, gheber@hdfgroup.org

false readings, data loss, and devices with rather limited capabilities. The visualization of the science application domain objects needs to be “conditioned” on the behavior of non-science application domain objects. Further complications arise from streaming data, for example, in online analysis. Where does one situate low-level details such as buffering? Should it reside in middleware glue code or be an intrinsic part of domain object behavior?

Topic 3.c. deals with extreme-scale data, or its cousin, extreme data reduction. Although visualization is a form of data reduction, perhaps not too many users would argue that visualization tools should perform the heavy lifting for the data reduction needed to cope with extreme-scale data. Reductions such as lossy compression [2] and multi-resolution representations, assisted by intelligent data buffering and “just-in-time data delivery”, have to occur much farther upstream and be part of domain object or domain object view behavior.

Topic 4. Deals with tools and technology to support equal access to relevant data and universally interpretable analysis. At the heart of the problem appears to be an overemphasis on code over data and domain objects. Scientists agree on mathematics and domain objects, but that agreement seems to end when visualization tools that can’t deal with domain objects get involved, and that creates the impression that it’s all about portable glue code and that data representations are mere technical details that can be made up ad hoc. We believe it is time to put domain objects and their behavior first, again, to achieve *data-driven behavior*. (Not to be interpreted as a choice or preference of a programming paradigm!)

Perhaps this quick selection gave at least the impression that there is a common thread to the limitations of state-of-the-art practice in visualization for scientific discovery, decision-making, & communication. That thread and threat (!) is the separation of data and behavior, which has led to visualization tools that are limited by their fundamental inability to deal with domain objects from a single application domain, let alone multiple domains in the same application. State-of-the-art visualization tools are ill-prepared for becoming the hubs capable of handling the explosion of new data sources and types, as well as unprecedented volumes of data. We believe there is an opportunity to pick up where research in scientific visualization stood before UI design and glue code integration became major concerns. With many more tools beyond object-orientation (as a programming paradigm) at our disposal and backed by advances in code generation, intelligent storage, and software-defined networking, a new breed of visualization tools that lets domain objects present themselves might be created over the next decade.

## References


**[1]** Wikipedia (2021, December 7), *Naked Objects.* https://en.wikipedia.org/wiki/Naked_objects

**[2]** Milan Klöwer et al., Compressing atmospheric data into its real information content. Nature Computational Science 1, pages 713–724 (2021) https://www.nature.com/articles/s43588-021-00156-2