Studies in History and Philosophy of Biological and Biomedical Sciences 44 (2013) 217–224

Contents lists available at SciVerse ScienceDirect

Studies in History and Philosophy of Biological and Biomedical Sciences

journal homepage: www.elsevier.com/locate/shpsc

Digital ‘faces’ of synthetic biology

Kathrin Friedrich

Academy of Media Arts Cologne, Peter-Welter-Platz 2, 50676 Köln, Germany

ARTICLE INFO

Article history:
Available online 8 April 2013

Keywords:
Computer-Aided Design
Software
Diagrams
Graphical User Interface
Visual Programming

ABSTRACT

In silicio design plays a fundamental role in the endeavour to synthesise biological systems. In particular, computer-aided design software enables users to manage the complexity of biological entities that is connected to their construction and reconfiguration. The software’s graphical user interface bridges the gap between the machine-readable data on the algorithmic subface of the computer and its human-amenable surface represented by standardised diagrammatic elements. Notations like the Systems Biology Graphical Notation (SBGN), together with interactive operations such as drag & drop, allow the user to visually design and simulate synthetic systems as ‘bio-algorithmic signs’. Finally, the digital programming process should be extended to the wet lab to manufacture the designed synthetic biological systems. By exploring the different ‘faces’ of synthetic biology, I argue that in particular computer-aided design (CAD) is pushing the idea to automatically produce *de novo* objects. Multifaceted software processes serve mutually aesthetic, epistemic and performative purposes by simultaneously black-boxing and bridging different data sources, experimental operations and community-wide standards. So far, synthetic biology is mainly a product of digital media technologies that structurally mimic the epistemological challenge to take both qualitative as well as quantitative aspects of biological systems into account in order to understand and produce new and functional entities.

1. Introduction

Synthetic Biology is a rapidly growing field in the context of biology, biotechnology, information science and engineering. On a basic research level scientists attempt to design and build biological components, networks and organisms *de novo*. But not only artificial systems should be constructed—the configuration of existing biological organisms and parts using engineering principles is also a branch of research. The *engineering paradigm* in synthetic biology promises that ‘biological machines are built from basic biomolecular components analogous to electrical devices, and the information flow among these components requires the augmentation of biological insight with the power of a formal approach to information management’ (Alterovitz, Muso, & Ramoni, 2009, p. 80). This endeavour demands, in particular, effective computational techniques to manage the complexity of biological systems[1] and therefore the large amount of data needed for their construction and reconfiguration (Purnick & Weiss, 2009, p. 410).

The project of ‘putting engineering into biology’ (Heinemann & Panke, 2006) is still taking place mainly within the borders of the computer’s dry lab (Merz, 2006), for scientists face problems of cell death, cellular noise, crosstalk and mutations when transferring components to *in vivo* environments, since ‘our biological knowledge and design capabilities are not yet at the level of sophistication needed for *a priori* design and production of a prototype with a fair shot at success’ (Alterovitz et al., 2009, p. 81, italics in original). However, the final goal of synthetic biology is not only to derive and integrate data from wet lab experiments but ultimately to manufacture the designs of the dry lab by wet lab practices such as automated assembly. In particular, computer-aided design (CAD) software programs like TinkerCell, BioJADE, CellDesigner and GenoCAD are intended to bridge the gaps between

---

E-mail address: kfriedrich@khm.de

[1] According to MacDonald et al., complexity ‘[…] can refer to both the specification of the structure of the system, i.e. constituent parts and their connections, and to the set of parameters describing its kinetics (production and decay rates for example)’ (MacDonald et al., 2011, p. 98).

1369-8486/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.shpsc.2013.03.017

wet and dry lab, user and data or algorithms and aesthetics by black-boxing and substituting quantitative with qualitative properties and vice versa.

TinkerCell, for example, enables researchers to virtually design biological systems, e.g. signalling pathways, with the help of prefigured diagrammatic symbols. The visually programmed diagrams cover the underlying mathematics within an appealing and easily operable graphical interface that is equipped with several experimental functions, e.g. stochastic simulation (MacDonald et al., 2011, p. 99). Ultimately, the 'a priori designs' ought to cut through directly from TinkerCell's visual surfaces to biological matter.To explore the *in silico* design of synthetic biological systems 'with a fair shot at success', I will focus on the role of computer-aided design software such as TinkerCell. In which respect does this software 'aid' design in synthetic biology by putting biological properties into operation as 'bio-algorithmic signs'? As I will argue, the epistemic and aesthetic challenge that is embodied in the software is to simultaneously black-box and bridge algorithmic and biological processes by and within visual surfaces. This paradox of simultaneously connecting and separating resides in digital media technology and from there encourages the visions of synthetic biology to become not just thinkable but (materially) operable.

2. 'Multi-linear ensembles': computer-aided design

As a specific computational technique, computer-aided design software is supposed to be a helpful instrument to design synthetic entities according to engineering principles, such as standardisation, modularity and automation (cf. Endler et al., 2009; Suarez, Rodrigo, Carrera, & Jaramillo, 2009). For example, Voigt stated in a recent publication that '[c]omputer-aided design (CAD) will become an increasingly important tool for synthetic biology, as designs become larger and more complex' (Voigt, 2011, p. xvii). Besides suggesting that CAD software is particularly able to manage the current challenges in synthetic biology, this statement treats CAD as merely a 'tool' for synthetic biology rather than a modality with its own means to generate knowledge and to carry out operations that literally matter.

The characterisations of CAD's purpose and its specific 'aid' for synthetic biology remain quite unclear throughout subject-specific publications. Ideally CAD should 'assist an engineer in the process of designing a system with a desired behaviour (specification and design) and understanding the system in sufficient detail to construct the physical realization of the system (assembly)' (Chandran, Bergmann, Sauro, & Densmore, 2011, p. 204). But since '[i]n biological engineering, there is no established design methodology for moving from a specification, or desired behaviour, to the end result, the living cell' (Chandran et al., 2011, p. 204), it is crucial to explore the currently deployed CAD programs. In so doing, it becomes obvious how CAD software generates and 'hosts' epistemic procedures and aesthetic features that mutually foster as well as limit the goals and practices of synthetic biology rather than just being a ready-to-hand tool. Unlike other CAD practices in engineering, such as machine construction, biological systems challenge the development of software applications with unknown parameters and the autopoiesis of genetic systems. The context of application of organisms is just as contingent as the system's behaviour within it. Hence, CAD in biology differs from other design practices since 'lifelike' algorithmic objects still lack the qualitative and contextual features of complex living systems. Whereas the parts, e.g. signalling or metabolic pathways, are about to be more sufficiently understood, it is hardly possible to assess the combination and the relations of the parts within a larger biological system. The quantification of biological properties does not yet overcome this lack of material and process-related qualities, but visual software interfaces in CAD seem to offer intuitive aesthetic mock-ups that cover the depth of uncertain biological and algorithmic processes through well-known and operable visuals.

Software in general is an integral part of digital media technologies as well as social, cultural and epistemic practices (e.g. Dodge & Kitchin, 2011, p. ix; Manovich, 2008). The more ubiquitous digital media technologies become in both everyday practices and in scientific contexts, the more 'invisible' the functions and properties of software seem to become. Attempts to describe the ontology or 'nature' of software without employing a specific example seem difficult (Dodge & Kitchin, 2011, pp. 23–45). Software is not 'an object' that can be grasped analytically; rather, it is a process that couples relations of and to other entities. Or, as David M. Berry states, '[s]oftware is a tangle, a knot, which ties together the physical and the ephemeral, the material and the ethereal, into a multi-linear ensemble that can be controlled and directed' (Berry, 2011, p. 3). Therefore, software—as a preliminary characterisation—needs to be approached as a relational process, which could be opened up and stopped at its subface, interface, and surface as well as on crystallisation points that frame the material, epistemic or aesthetic conventions of this process. In so doing, one might get closer to 'the stuff of software in some of the many ways that it exists, in which it is experienced and thought through, and to show, by the interplay of concrete examples and multiple kinds of accounts, the condition of possibility that software establishes' (Fuller, 2008, pp. 1–2).

Software programs get even more 'tangly' and seemingly loaded with 'possibilities' when they are designated to aid design. Design is a crucial and at the same time blurry term used in synthetic biology (Mackenzie, 2010, p. 181). This mirrors how design as a practice is employed by synthetic biologists themselves, namely as a process that 'is as much an exploratory process as it is a rational design process' (Chandran et al., 2011, p. 203). Hence, Adrian Mackenzie's notion of design as a meta-technique provides a starting point to think about software-aided design as 'a meta-technique in that it organises, groups, assembles and subsumes other techniques, practices, methods, protocols, knowledges, services, and infrastructures into specific arrangements, while at the same time, appearing to stand outside them' (Mackenzie, 2010, p. 183).

As I will show, software is a powerful aid for design in synthetic biology in that it establishes relations between computability, visibility and materiality and simultaneously bridges and black-boxes these relations by its own relational character. Hence, cognition and knowledge production cannot be analysed separately as 'epistemic thinking' on the one hand, and 'aesthetic doing' on the other (Krämer, 2011, pp. 280–281); in CAD both engender each other in embodied and embedded practices. In that software establishes 'a multi-linear ensemble that can be controlled', it reflects that design in synthetic biology oscillates between the numerical determination of computational models and the exploratory uncertainty that is visualised within the graphical user interface (GUI) of programs like TinkerCell.

3. 'Intricate interplays' in synthetic biology: CAD software

Computer-aided design programs like TinkerCell (previously called Athena) are intended to enable the visual programming of, e.g. biological and biochemical networks, and 'to bridge the gap between computational modelling and biological data' (Chandran, Bergmann, & Sauro, 2009). TinkerCell is a free, open-source software for circuit design, developed at the University of Washington's Bioengineering Department. By the application of TinkerCell '[u]sers can construct biological networks using genes, promoters, proteins, cells, etc. and analyze the network through simulations or other available functions. Each item has data associated with the