Marcel Borowski (2021): Exploring Possible Futures With Computational Media. In: Proceedings of the 19th European Conference on Computer-Supported Cooperative Work: The International Venue on Practice-centred Computing on the Design of Cooperation Technologies - Doctoral Colloquium Papers, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/ecscw2021_dc05

Exploring Possible Futures With Computational Media

Marcel Borowski
Department of Computer Science, Aarhus University, Denmark
marcel.borowski@cs.au.dk

Abstract. When thinking about software, we often think about applications with a fixed set of functionality that are tied to particular devices. While applications provide functionality for their use cases, e.g., a word processor for writing text, they are usually inflexible in providing functionality beyond their often narrow use case. An alternative model considers software as a medium — Kay and Goldberg think of it as expressive and malleable as paper or clay. My research explores computational media and possible futures of software built with computational media by following an explorative and qualitative research approach using high-fidelity prototypes. I present preliminary results of properties of computational media, the potentials and limits of them, and use cases where software built with computational media excels. With my research, I aim to contribute to rethinking applications as the predominant model of software, shifting software towards a more malleable foundation, overcoming the boundaries of applications and devices.

1 Background and Motivation

We often think about software as monolithic applications on particular devices: We use our laptop to write text in Microsoft Word or edit photos in Adobe Photoshop, and use our tablet with pen input to read and annotate documents in Adobe Acrobat Reader. By doing so, applications compartmentalize functionality into fixed silos — applications are good at doing what they do, but going beyond that, they are often highly inflexible. The notion of computational media looks at software from a different angle: instead of seeing it as tool for a specific task, it considers software as a malleable medium, that users can form and mold to their needs and easily share with others. Kay and Goldberg (1977) compare computational media with paper or clay, allowing for many different ways of using it. While computational media has a long history with early platforms such as the Dynabook (Kay and Goldberg, 1977) and Boxer (diSessa and Abelson, 1986), and experienced a recent resurgence in new platforms like Webstrates (Klokmose et al., 2015), its potentials and limitations are mostly unexplored.

My notion of computational media builds on the idea of *shareable dynamic media* by Klokmose et al. (2015): a medium that is characterized by the three properties *shareability* (being shareable with other users and allowing for collaboration), *distributability* (being distributable across heterogeneous devices), and *malleability* (being extendable and reprogrammable). Building on Webstrates, Rädle et al. (2017) created Codestrates, which added an authoring environment that "blurs the distinction between use and development of applications" (Rädle et al., 2017). It inherits the properties of Webstrates and implicitly adds the forth property *computability* (being able to edit and execute custom computations within the document). These four properties are the foundation for my proposed properties of computational media.

2 Research Goals

My PhD project aims to explore the use of computational media as malleable software in different scenarios. The aims are threefold and interrelated:

RG1 — Defining Properties of Computational Media. First, my project aims to define properties of computational media in more detail. It will build on existing work (Klokmose et al., 2015; Rädle et al., 2017) and discuss properties in the process of different cases.

RG2 — Exploring Potentials and Limits of Computational Media. Next, my project will explore and uncover potentials and limits of computational media. These potentials and limits will be related to the defined properties to understand how they influence computational media.

RG3 — Identifying Domains and Use Cases for Computational Media. Lastly, my project aims to identify domains and uses cases in which computational media is well suited to be used. This aim, again, is related to the preceding aims and allows for a better assessment of the capabilities of computational media.

3 Research Approach

My project aims are rooted in exploratory research, therefore, my research approach focuses on qualitative methods and exploration rather than comparative lab studies. Computational media as software is radically different from the established model of turn-key applications found in most commercial software products. Thus, my project aims to explore and uncover *possible futures* (Salovaara et al., 2017) of software built with computational media. To operationalize this aim, the project utilizes high-fidelity prototypes and participatory design (Bødker et al., 1995). The prototypes act as *computational alternatives*: "manifestations of research and design ideas as well as demonstrations of possible ways to move ahead" (Korsgaard et al., 2016). Using high-fidelity prototypes allows users to imagine in more detail how software built with computational media is different from classic applications. The development of these prototypes is supported by participatory design and co-design methods (Bødker et al., 1995; Sanders and Stappers, 2008). Letting users participate in the design process allows them to shape software to their needs while reflecting on how computational media, rather than applications, contributes to achieving these goals. Working closely with participants in such a design process, further, provides insights into their understanding of software as a computational medium.

4 Preliminary Results

So far, I worked on two projects: (1) Deploying a computational medium in a nanoscience lab (Nouwens et al., 2020) and (2) using computational media in a co-design study about collaborative writing (Larsen-Ledet and Borowski, 2020; Borowski and Larsen-Ledet, 2021). I will briefly introduce these projects and then summarize the results of these two projects in relation to my research goals.

Computational Media in a Nanoscience Lab. My first project involved deploying a computational media prototype, the *computational labbook*, in a study with biomolecular nanoscientists. The work of these scientists relies on computational tools for designing RNA structures in the field of RNA origami. The prototype was developed iteratively together with the scientists in a participatory design process. The labbook acted as a computational alternative (Korsgaard et al., 2016) to co-create possible futures (Salovaara et al., 2017) of a computational laboratory notebook (Nouwens et al., 2020). This project provided insights to deepen the properties of computational media and their facets, and how they relate to the case of computational nanoscience.

Computational Media in Co-design Workshops. My second project focused on using a computational media prototype in a co-design workshop to facilitate cooperative prototyping (Bødker and Grønbæk, 1992). The prototype was malleable and allowed participants to extend and reprogram it during the workshop. This work provided insights into the difficulty of making use of malleability and reprogrammability as an end-user. It, further, showed the benefits of employing computational media in prototyping workshops during a co-designs study and limitations of current platforms like Codestrates (Rädle et al., 2017).

4.1 RG1 — Properties of Computational Media

The first three properties are defined by Klokmose et al. (2015), while the fourth property was derived from Rädle et al. (2017) during my first project (Nouwens et al., 2020). The facets of the properties below are based on my two projects if not cited otherwise.

Distributability. Computational media is distributable in three ways: documents, functionality, and computation. *Documents* of computational media are distributable beyond device and operating system borders. *Functionality* of a document, e.g., parts of the user interface, can be distributed across multiple devices, allowing for cross-device interaction and a fluid rearrangement of parts of a document or the user interface on multiple devices. Lastly, *computation* can be distributed across devices and servers, providing means to offload computation to more powerful client devices, e.g., from a phone to a desktop computer (Badam et al., 2018), or to a more powerful server (Klokmose et al., 2019; Nouwens et al., 2020).

Shareability. Computational media is shareable, allows for collaboration, and enables users to more easily receive help from peers. Documents are *shareable* with other users, they are held in common and can be accessed by multiple users simultaneously. In Webstrates, for example, documents are shareable using links. Documents are also synchronized and automatically updated, enabling real-time *collaboration*. Finally, shareability enables users not only to collaborate together to solve a task, but also to *receive or give support* to others. This was employed by nanoscientists in the first case to receive help from remote programmers (Nouwens et al., 2020).

Malleability. Computational media is malleable and supports multiple levels of tailoring: extensibility, configurability, and reprogrammability.[1] *Extensibility* allows users to extend and reduce the functionality of a document, e.g., by using plug-ins that can be added or removed. *Configurability* enables users to customize existing functionality, e.g., by editing parameters without needing to change the underlying code. *Reprogrammability*, lastly, provides tools to view, edit and re-run

[1] These levels are inspired by the levels of end-user tailoring by Mørch (1997).

the underlying code of functionality from within the document, i.e., documents are self-contained and inhabit the code of their functionality.

Computability. Computational media is computable, allows custom code execution, and contains the computational environment within itself. *Code execution* of arbitrary code is possible from within documents and outputs of computations can be stored and presented directly within the document. Ideally, computation should not be restricted to one programming language but support polyglot language interoperability, allowing to combine multiple programming languages in the same document. The *computational environment* is part of the document and independent from devices, i.e., code executions behave the same on all devices.

4.2 RG2 — Potentials and Limits of Computational Media

Distributability. Unsurprisingly, distributability of documents is beneficial when users use multiple devices at multiple locations. Systems like Google Docs employ such a way of distribution already. While cloud storage like Dropbox synchronizes files each device still needs to be able to open these files, requiring to install particular applications. A limitation of this is the need to be connected with a server while working on documents. While the possibility of cross-device interaction is beneficial, there are limitations like a *legacy bias* (Plank et al., 2017) towards using only one device at a time — like in my first project (Nouwens et al., 2020).

Shareability. With documents being held in common, shareability provides means for both synchronous and asynchronous collaboration. As a computational platform, this is not restricted to, e.g., text as in Google Docs but also enables for collaborative programming practices. This can, for instance, be useful when working on programming assignments together (Borowski et al., 2020), when requiring help from a more capable peer (Nouwens et al., 2020), and when prototyping (Borowski and Larsen-Ledet, 2021). While documents held in common are useful in these cases, it also poses potential risks regarding security and privacy.

Malleability. Malleability is beneficial, as it allows users to customize software on a visual *and* functional level. This can be useful in scenarios like prototyping (Borowski and Larsen-Ledet, 2021) and allows users to mix and match functionality (Borowski et al., 2018). Being a self-contained system with easy access to code is needed in such a case. Often, however, this still requires programming skills on the side of the user, which can provoke new problems, as users might feel disempowered by their software (Nouwens et al., 2020). Too much choice in what tools one wants to use can overwhelm users (Larsen-Ledet and Borowski, 2020).

Computability. Computability provides the foundation for leveraging computation in more areas of software. It allows to “opportunistically use scripts from different sources, written in multiple languages" (Nouwens et al., 2020). Bundling the computational environment with the document and not relying on specific devices or dependencies, such as local Python distributions, allows sharing documents without worrying about how a document looks or behaves on different devices. But this bundling can also confuse as it removes the familiar separation of functionality in applications and content in files (Borowski and Larsen-Ledet, 2021).

4.3 RG3 — Use cases for Computational Media

Frontiers of Science. The first domain to apply computational media are "frontiers of science," i.e., fields where there are no established applications available — either because it is not profitable for companies to develop them or because the field is just emerging. Computational media can provide users "with the flexibility of scripts together with the accessibility of applications" (Nouwens et al., 2020), placing itself in between the two. This is mainly supported by the properties malleability and computability.

Prototyping and Co-Design. As shown in my second project (Borowski and Larsen-Ledet, 2021), another domain that can be supported by computational media is cooperative prototyping (Bødker and Grønbæk, 1992) in participatory design and co-design processes. Being self-contained and malleable, documents in computational media can quickly be changed and iterated. In addition, it allows also participants to join in the design process, being able to modify the functionality of documents without the need for additional development environments, which are otherwise often required to modify software.

Collaboration. The distributable and shareable nature of computational media makes it a good foundation for collaborative use cases. For example, collaborative data exploration (Badam et al., 2018), video editing (Klokmose et al., 2019), programming (Borowski et al., 2020), prototyping (Borowski and Larsen-Ledet, 2021). Malleability can, further, help to support idiosyncratic needs (Larsen-Ledet and Borowski, 2020) by letting users tailor their functionality.

5 Next Steps

In the second half of my PhD, I focus on the property malleability: A common problem in both projects was the difficulty of users to make use of malleability. Although the prototypes in both studies were self-contained and entirely reprogrammable, it was difficult for participants to make use of it. This was both due to a lack of programming skills and the difficulty to understand existing code in the system. To support users without or with little programming skills to make use of malleability, I plan to build a new prototype that follows a declarative programming model to define interaction, making it easier to understand the existing code/functionality and to extend and reprogram it. With this prototype, I then plan to conduct another co-design workshop or hackathon — possibly again in the domain of collaborative writing — to deepen the initial insights into using computational media in prototyping workshops and provide answers about how a declarative programming model can support the accessibility of malleability.

6 Expected Contribution

The project's outcome will be related to the project goals: (1) a definition of properties of computational media, (2) potentials and limits of these properties and computational media overall, and (3) domains and use cases for computational media. Furthermore, the project will produce several high-fidelity prototypes demonstrating possible futures of software building on computational media and open new possibilities for future research by providing a technical foundation as well as guidelines for the design of computational media platforms and how they can complement current turn-key applications. These new insights and technical foundations are intended to shift the focus of software towards a common, shareable, and malleable foundation, in order to overcome the boundaries of applications and devices.

Acknowledgments

I thank Susanne Bødker and Clemens N. Klokmose for their mentorship and guidance, Ida Larsen-Ledet for providing feedback, and my project collaborators and co-authors. My project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 740548).

References

Badam, S. K., A. Mathisen, R. Rädle, C. N. Klokmose, and N. Elmqvist (2018): 'Vistrates: A Component Model for Ubiquitous Analytics'. *IEEE Transactions on Visualization and Computer Graphics*.

Bødker, S. and K. Grønbæk (1992): *Design in Action: From Prototyping by Demonstration to Cooperative Prototyping*.

Bødker, S., K. Grønbæk, and M. Kyng (1995): 'Cooperative Design: Techniques and Experiences From the Scandinavian Scene'. In: *Readings in Human–Computer Interaction*.

Borowski, M. and I. Larsen-Ledet (2021): 'Lessons Learned From Using Reprogrammable Prototypes With End-User Developers'. In: *Proceedings of the 8th International Symposium on End-User Development*. In Press.

Borowski, M., R. Rädle, and C. N. Klokmose (2018): 'Codestrate Packages: An Alternative to "One-Size-Fits-All" Software'. In: *Proceedings of the 2018 CHI Conference Extended Abstracts on Human Factors in Computing Systems*.

Borowski, M., J. Zagermann, C. N. Klokmose, H. Reiterer, and R. Rädle (2020): 'Exploring the Benefits and Barriers of Using Computational Notebooks for Collaborative Programming Assignments'. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*.

diSessa, A. A. and H. Abelson (1986): 'Boxer: A Reconstructible Computational Medium'. *Communications of the ACM*.

Kay, A. and A. Goldberg (1977): 'Personal Dynamic Media'. *The Computer Journal*.

Klokmose, C. N., J. R. Eagan, S. Baader, W. E. Mackay, and M. Beaudouin-Lafon (2015): 'Webstrates: Shareable Dynamic Media'. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*.

Klokmose, C. N., C. Remy, J. B. Kristensen, R. Bagge, M. Beaudouin-Lafon, and W. Mackay (2019): 'Videostrates: Collaborative, Distributed and Programmable Video Manipulation'. In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*.

Korsgaard, H., C. N. Klokmose, and S. Bødker (2016): 'Computational alternatives in participatory design - Putting the T back in Socio-Technical research'. In: *Proceedings of the 14th Participatory Design Conference: Full Papers - Volume 1*.

Larsen-Ledet, I. and M. Borowski (2020): '"It Looks Like You Don't Agree": Idiosyncratic Practices and Preferences in Collaborative Writing'. In: *Proceedings of the 32nd Australian Conference on Human-Computer Interaction*.

Mørch, A. (1997): 'Three Levels of End-User Tailoring: Customization, Integration, and Extension'. In: *Computers and Design in Context*.

Nouwens, M., M. Borowski, B. Fog, and C. N. Klokmose (2020): 'Between Scripts and Applications: Computational Media for the Frontier of Nanoscience'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*.

Plank, T., H.-C. Jetter, R. Rädle, C. N. Klokmose, T. Luger, and H. Reiterer (2017): 'Is Two Enough?! Studying Benefits, Barriers, and Biases of Multi-Tablet Use for Collaborative Visualization'. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*.

Rädle, R., M. Nouwens, K. Antonsen, J. R. Eagan, and C. N. Klokmose (2017): 'Codestrates: Literate Computing with Webstrates'. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*.

Salovaara, A., A. Oulasvirta, and G. Jacucci (2017): 'Evaluation of Prototypes and the Problem of Possible Futures'. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*.

Sanders, E. B.-N. and P. J. Stappers (2008): 'Co-creation and the new landscapes of design'. *CoDesign*.