medienhaus/ separating data and design

Authors Marcel Haupt, Andi Rueckel, Robert Schnüll, Florian Weber (udk/spaces team der UdK Berlin)

Context InKüLe - Innovationen für die Künstlerische Lehre UdK Berlin/www.inkuele.de

Date April 2023

Abstract

When designing digital tools, it's important to consider the concept of separating data and design; one of the most fundamental principles in software development. But why is it so crucial and how does it relate to artistic education? In this text, we will delve into the benefits of separating data and design, opening access to the distinct layers, and explore how the "Separation of Concerns" intersects with artistic education. We hope to highlight the importance of creating digital tools that are not only functional, but also try to meet the different forms and needs within artistic education, as well as the disciplines' unique teaching methods and learning styles.

Separation of Concerns

Before we can talk about the benefits of this principle and its application in artistic education, we need to first provide a short overview of the principle itself.

Separating data and design is a fundamental concept in computer science and known as "Separation of Concerns" (SoC). It can be applied to both the architectural layer of the software, and the programming level. We will mainly focus on the architectural aspect of the concept in this paper, but want to acknowledge its equal importance on a programming level. The idea behind SoC is to separate different concerns or areas of functionality, such as data management, user interface, business logic and system infrastructure, into independent modules. Each module usually has a clearly defined responsibility and its implementation should be independent of the other modules. By doing this, changes to one module do not affect the others and the system as a whole can be easily modified or extended. It also allows developers to work independently on different aspects of the system. They can make changes, identify problems and fix bugs more quickly, promoting modularity, reducing complexity, and enhancing maintainability, flexibility and scalability.

For example, in web development, SoC can be achieved by separating the following layers:

- **the presentation layer**, which includes HTML, CSS, and JavaScript, i.e., the public website.

- **the application logic**, which is the server-side code responsible for tasks such as user authentication during login.

- **the data layer**, such as the database containing user account information.

By making sure there is as little overlap as possible between these different concerns, developers can better manage and maintain the application's codebase. Not only making it more modular, but also making it far easier for new code maintainers to understand and work with the code in the future.

The problem with platform economies

Disclaimer: As we look at the importance of keeping data and design separate, we'll be taking a closer look at platform economies as an example of what can happen when this principle is ignored. It's important to note that on a technical level, most (if not all) of the major tech companies likely adhere to the "Separation of Concerns" principle. However, as their code bases remain closed source, we cannot verify this at the time of writing. Despite the separation potentially being present behind the scenes, it is not typically noticeable to users. They do not have direct access to their account data and instead interact with it through the platform's interface. This lack of transparency and control over personal data highlights the importance of separating data and design, as well as the necessity for unrestricted, open access to this data, all of which can ultimately benefit the end-user.

Digital communication and digital tools are an essential aspect of our lives and studies. The COVID 19 pandemic has reinforced and amplified this. Many institutions and universities are forced to rely on platforms to stay in touch with their students and faculty. While these platforms have proven to be a valuable tool in maintaining contact with students, there are a myriad of downsides. One of which being the dissonance between public image and the actual business behind the brand. Most of these platforms rely heavily on a sense of community. While they are actually quite good at conveying this, it's easy to get the impression that this isn't actually their priority.
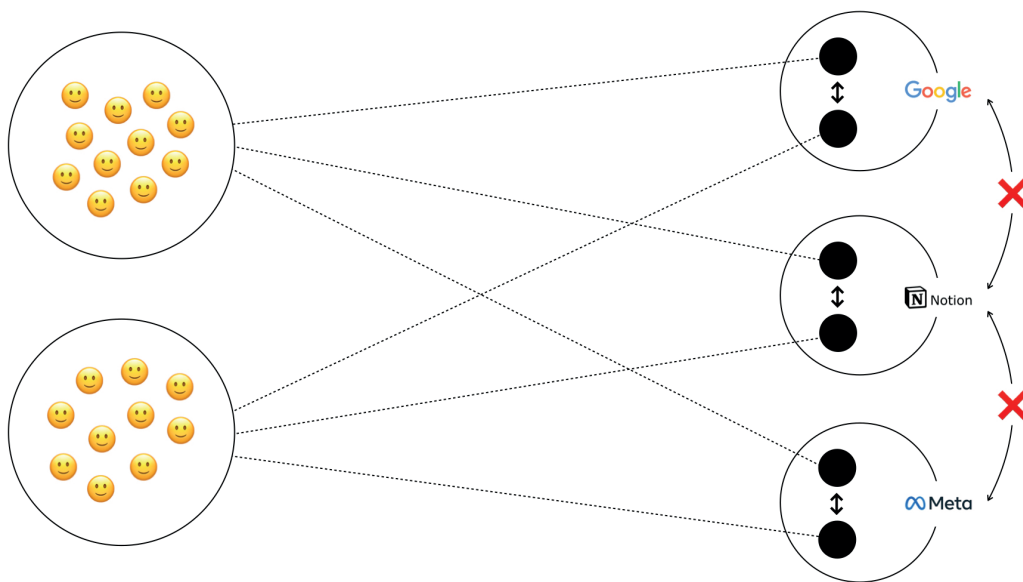
Let's say, for example, we have a community which uses Google Drive or Google Docs, they they use Notion for collaborative writing, and Instagram and Facebook to communicate with the rest of the world. Everything seems to work great within their respective bubbles, and they can exchange data and communicate with each other within each

individual platform. However cracks begin to appear when these platforms try to communicate with someone other than themselves, for the simple fact that they are not interoperable. This is not by chance, but by design. By intentionally prohibiting users from exchanging data across platforms, they have created an environment that discourages cross-platform collaboration. They want users to spend as much time as possible on their own platform, and while "share buttons" are a common feature, they appear to be primarily designed to drive traffic away from one platform to another.

Despite trying to appear to be fostering open communities, these platforms are closed systems and, at their core, data-driven companies. They collect and store vast amounts of user data, which they use for their own purposes. As a result, content in various apps and tools are mostly locked within their servers (more often than not somewhere in North America). This means that control over data is not in the hands of users. It's not even in the hands of potential system administrators of the institution or university where you might be working or studying. Rather, it is in the hands of big-tech corporations perpetuating surveillance capitalism with users themselves having little to no control over how their data is used. While awareness for this has grown, most are often unaware of or underestimate how and where their data is being collected, stored, and shared.

content is now **»locked«**
inside different apps/tools,
inside **«specific»** platforms.

SoC in art education
(Data) Sustainability

The point of sustainability, or rather the lack of sustainability, is a crucial aspect to consider in relation to our data. Please note that in this section, we are not talking about environmental sustainability, but about the longevity and accessibility of data. While the climate crisis is the defining crisis of our time, the scope and complexity of this issue is beyond the scope of this article.

In discussions about technology, we often emphasise the potential benefits of using the latest technological advancements, and often forget the present state of our data and content. What happens if this data suddenly becomes unavailable? When platforms cease to exist or, potentially on a whim, decide to restrict access to their platform? Accessing data that was stored with them would be challenging to say the least. This is because, as we already discussed, data is typically locked within the platform's proprietary system, making it difficult to retrieve without access to the platform itself and creating a single point of failure. It is critical to consider accessibility to the data when choosing technological solutions, and to ensure that backups and other measures are in place to protect valuable data in case of potential platform closure or other disruptions such as hardware failure, cyber attacks, and natural disasters.

As an institution, as artists, designers, and as educators, we have a responsibility to think critically about the infrastructure we use and deploy. We need to ask ourselves where our data is stored and who should make this decision to begin with. It is necessary to consider the creators of the spaces, interfaces, and rules that we work with. We frequently use generic interfaces that were never actually designed with artistic education in mind. Furthermore, the fact that most of these platforms are designed with a different purpose in mind altogether — the collection and analysis of user data — means that they do not always prioritise user experience. In other words, design decisions are made based on what is best for the company in a market economy, not what is best for the user. This is impractical at best and can even be harmful at worst.

scope of action is dictated
by the interface,
and **the interface is dictated**
by the platform.

https://zengarden.com
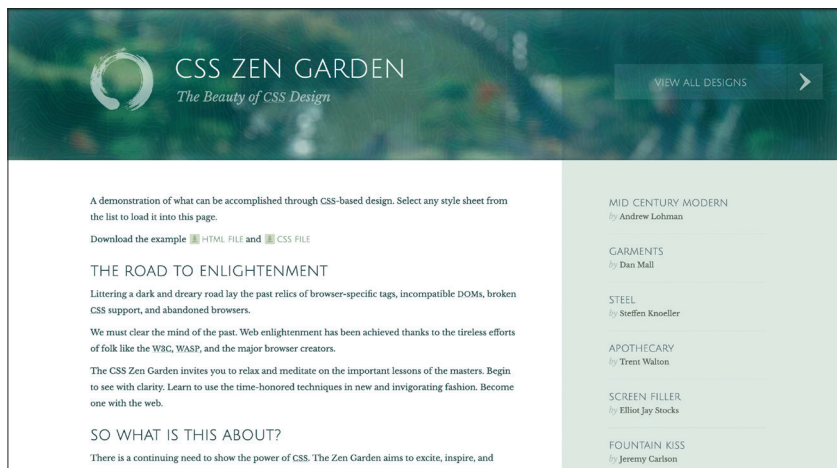
Adaptability

SoC has several advantages, one of them being the adaptability it introduces to a system. As already mentioned previously it enables developers to modify one aspect of a system without affecting the others. A great visual illustration of this is the CSS Zen Garden website. It demonstrates the concept beautifully in web design, where completely separating data and the visualisation of this data is a generally accepted "best practice".

The idea behind the website is to showcase how the same HTML code (the data) can be styled in different ways, using only CSS (visualisation). The website features a single HTML file, which remains constant across all designs. However, the CSS code is entirely different for each design, creating a unique look for each submission.

Rundgang and CMS

Another example of this comes from our own research and work. For the digital representation of the Rundgang (Open Days at Berlin University of the Arts), the **udk/spaces** team realised a system with the SoC principles in mind. We created a custom content management system (CMS) for the students and educators of the university based on the [matrix] protocol. Like any other CMS, it allowed users to to create, edit, and publish content on a publicly accessible website: https://rundgang.udk-berlin.de/

While, at first glance, it seems like significant added effort, writing our own CMS comes with many advantages which outweigh the extra effort required compared to using pre-existing solutions. One of them being that we could ensure that published content elements are all individually accessible to other users via an API[1]. This means that any element included in a content published through the CMS, such as a single paragraph of text, an image file or an audio file, can be accessed and used separately in other applications.

---

1 An API (Application Programming Interface) is a way for two computer programs to communicate with each other. It acts as a middleman between to programs, allowing them to talk to each other and share information. Not unlike ancient messengers (preachers) carrying messages between two states. Every time we use apps on our phone, the app uses an API (our messenger) to communicate with a server, sending information to the server. The server then performs an action with this information and sends the new data back to your phone.
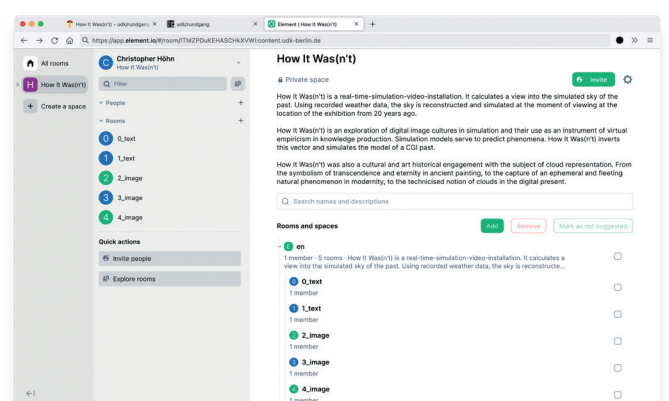
In order to netter understand this we will take a look at three examples which all use the same data source. Yet each example shows the data differently, just like zengarden.com.
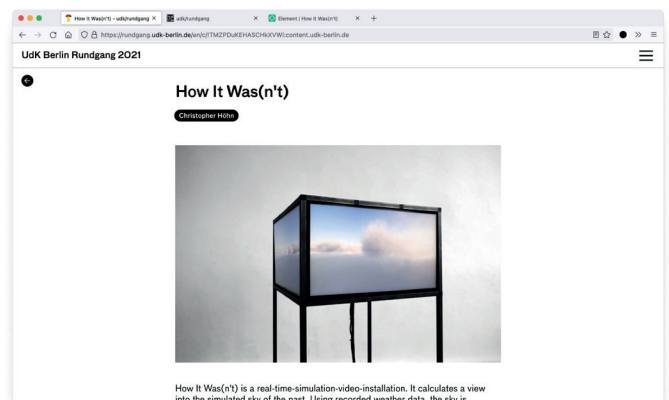
rundgang.udk-berlin.de public website
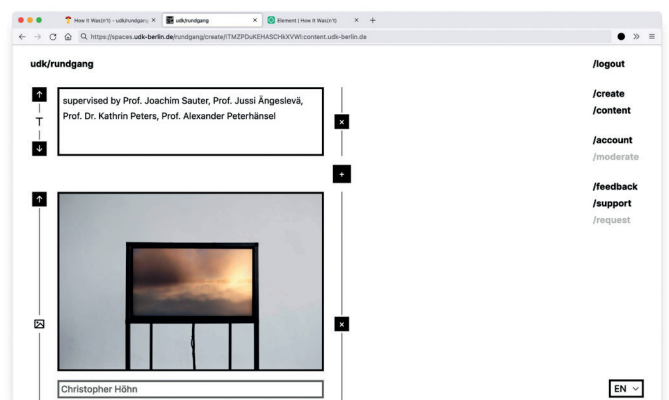
udk/rundgang content management system

Element-Web general purpose [matrix] client

In contrast to zengarden.com, we now have three completely different applications visualising this data. A normal website, a content management system, and a chat client – each presenting the same data in their own unique way.

You might ask, what is the point of viewing an art project on a chat client? Frankly, not much. But it does demonstrate that this data can be accessed and independently viewed in multiple ways. It opens up the opportunity for students and other users to create their own interfaces, art projects, or simply use the data to display it on their own websites. And, on top of that, it automatically creates an archive of every Rundgang which is independent of the specific design of the year it occurred.

Archives

An additional benefit of the separation of concerns principle, which is particularly interesting for universities and other educational institutions, is the (almost) automatic archiving of data. But before we can go further into the benefits of this we need to understand the difference between an archive and a database.

Archives and databases share a lot of similarities and even though it is fair to say that databases are archival in nature, not every database can automatically be considered an archive. One of the biggest differences is the purpose they serve: archives store historical data that is no longer actively used, while a database stores data which still is. Yet, the transition from database to archive can be a quick one, being as easy as generating a backup of the database, ideally storing it in a separate location. Some modern databases blur the lines between the two even further. They're built-in versioning tools enable the database to act as both an archive and database at the same time.

By separating data and design elements, the content created by educators and students can be stored and organised in an accessible manner, independent from its visualisation. This not only makes it easier to retrieve information as needed but also helps to ensure that valuable data is not lost over time and preserved for future generations. An important aspect of archiving is making information accessible to those who need it. We must therefore ensure that this data is not gatekept from the users by making (published) data and the resulting archives accessible to them, encouraging remix culture and participation in the process. When data and design are separated, it becomes easier to remix and reuse content, as each component can be easily accessed and modified. The data in these archives can become a new source of inspiration and discovery which can encourage participation and collaboration. Artists and designers are able to work independently or together to create new and potentially innovative content.

Data as a source of inspiration

By making the published data openly accessible to the students and educators, they are able to use it in their own artistic practice. The data can be a source of inspiration for creating new works of art. Connections and relationships between the data can be ex- plored, revealing patterns and links that may not be apparent at first glance. Artists and designers can also be inspired, or inspire us to explore the data in different ways.

Two such example are the works of Bruno Gola and Anastasia Almosova.

Bruno Gola used all the works that were created and uploaded for the 2021 Rundgang in his work "Rundgänger" to create an autonomous digital exhibition guide accompanying the Open Days at Berlin University of the Arts.
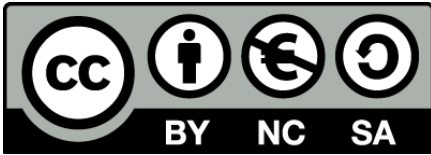
https://bgo.la/rundgaenger/

Anastasia Almosova utilised the same data to examine digital spatiality. She created an immersive environment to explore the students' work in 3D space, experimenting with non-linear narration and transgressing the limits and boundaries of physical exhibitions.

https://anastasiaalmosova.com/

Conclusion

As we rely more and more on digital tools for communication and education, it is impor- tant to keep transparency, open access to data and the need for interoperability in mind when choosing or designing these tools.

In artistic practice, we constantly recombine, inspire, and draw on various content, both consciously and unconsciously. By utilising the SoC approach and providing open acces- sibility to data, we can try to create digital tools which meet the unique needs of differ- ent disciplines and learning styles, while also being functional, sustainable, and empowering for users. It provides an opportunity for participation, allowing for the creation of new projects, perspectives, and perhaps even collaborations. Users within these sys- tems are given a choice, they can create their own interfaces, build applications, or use the data for their own work. This level of variability is a key principle being pursued by the udk/spaces team, working to unite these principles within udk/spaces in the future.