Aalborg Universitet

Interfering with the black-box-tradeoff model

Gephisto, a one-click Gephi for critical technical practice

Jacomy, Mathieu; Munk, Anders Kristian

Link to publication from Aalborg University

Interfering with the black-box-tradeoff model: Gephisto, a one-click Gephi for critical technical practice

Abstract

This paper presents and justifies Gephisto, an experimental tool visualizing networks in one click. Gephisto's design exemplifies how we can interfere with a user's utilitarian goals, by giving them what they wish (an easy way to get a network map) but in disobedient ways (the produced map is different every time the tool is used) that encourage them to engage further with the tool's methodological tenets. As an apparatus, Gephisto aims to incentivize untrained users to become more critical of their network mapping practices. As an intervention into the field of digital methods, it aims to show that tools that support critical thinking do not have to be hard to use and hostile to beginners. We criticize the idea that tools range from easy-to-use black boxes for unreflexive lazy-thinkers, to complex and demanding instruments for hard-thinking experts. We argue that learners need ease of use and critical thinking at the same time, and that it is possible to design tools that support both needs at once. We offer an alternative model where we acknowledge the active role of the user in deciding the tradeoff between learning to master the tool, and progressing toward their utilitarian goals. We argue that the design of the tool should not oppose the beginner's need for assistance in decision making, but find other ways to incentivize critical thinking.

Introduction

It is a common refrain in critiques of computational tools for digital methods research that ease of use dissuades critical thinking. Perhaps surprisingly, this also seems to be the case for relatively complex tools like the visual network analysis platform Gephi. Although the tool was designed specifically with the problem of critical thinking in mind - it has no default settings and therefore forces its users to make active choices wherever possible - it has still seen a use practice develop around it which stabilizes certain technical decisions as a *de facto* default and thus enables a fairly uncritical and easy-to-use style of network visualization to proliferate. In short: despite the best intentions of the tool designers to promote critical technical practice (CTP) by encouraging experimentation and exploration, it all comes down to whether the user is seeking that complexity or actively trying to avoid it. If the latter is the case, then ease of use can be achieved through the distributed network of tutorials, case examples, recipes, and support forums that are gradually put in place around the tool by its users. The complexity of Gephi is not seen as a potential for CTP by this group of users, but as a nuisance that must and can be overcome.

In this paper we therefore experiment with a radically different approach to the design of a visual network analysis tool for CTP. Taught by the example of Gephi we will no longer presume that the complexity of a tool can prompt unwilling users to slow down reasoning. On the contrary, we will presume that users who are not already looking for CTP will always find ways to avoid it in order to get what they want, in this case nice visual networks in a few prescribed steps. This leads us to propose a tool designed around a Faustian bargain in which the users who are seeking to avoid complexity get exactly what they want. As in all deals with the Devil, however, there is a catch. We call the tool Gephisto.

Gephisto

Get a **network map** in one click.

(but it's different each time)

Drop your GEXF here
or click to upload

Or try a sample network:

Europe   C. Elegans   Migrations

Figure 1: Landing page of Gephisto.

Gephisto promises to produce a network map in one click. It is available as a web app (fig. 1) and does not require any installation. The user inputs a network file, the tool generates a map, complete with a legend, which is then made available for download. Gephisto always tries to make the best network maps possible, but even so, there are *a priori* many equally valid ways to visualize the same graph. Without contextual knowledge of how the visualization is going to be used or what kind of narratives it is going to support, these choices are basically arbitrary. Indeed, this contextual knowledge typically emerges or becomes clearer in dialogue with the network visualization as the user experiments with different settings. This is why the original Gephi was designed with no default settings in order to prompt such experiments. In Gephisto's case, however, no dialogic effort or prior knowledge of these arbitrary choices are required of the user. Instead, since choices are equality valid without contextual information, Gephisto makes the choices at random and simply supplies a map when demanded to do so. The user, however, is offered the possibility of generating a new map for their input network as many times as they want. Gephisto always complies, but makes sure to never use the same settings

twice. In fact, it will actively try to maximize the variation in the output visualizations departing from its random starting point (fig. 2).
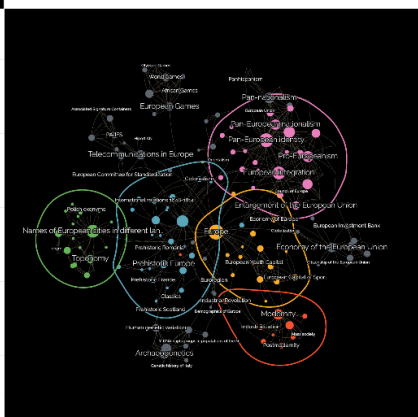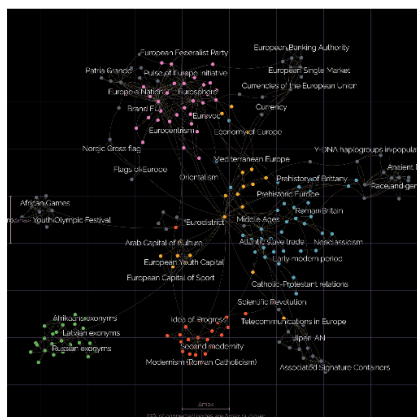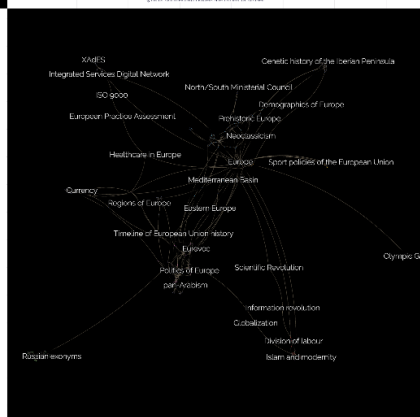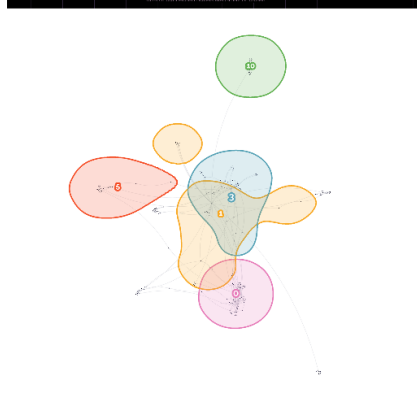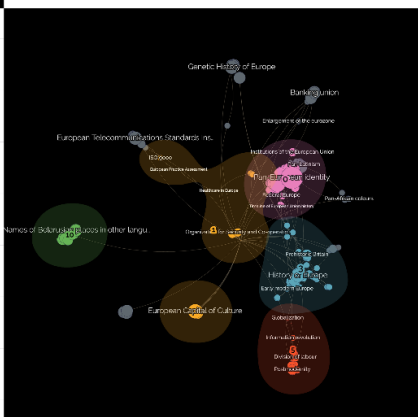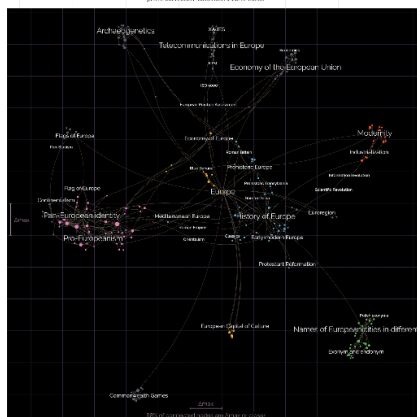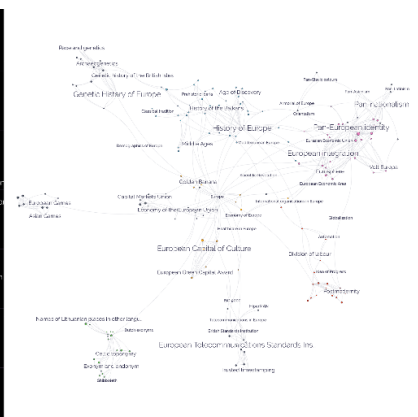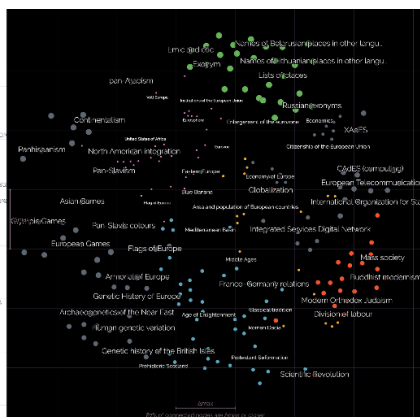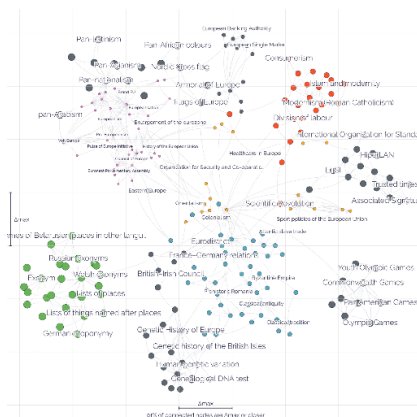
With this design, we aim at engaging the user into thinking about what making a network visualization entails, despite the apparent premise that it requires little critical reflection. Indeed, the key to our argument, and to Gephisto's duplicity, is how the allure of ease of use and blackboxing can be weaponized to promote their own opposites. Taking inspiration from the original Mephistopheles, we experimentally turn CTP into a matter of trickery.

The black box tradeoff

In this section we retrace the idea, found in the literature on tool criticism, that there is a *black box tradeoff*: either you design for ease of use, which entails hiding away complexity, alleviating the user of difficult choices but also dissuading critical thinking, or you promote prompt reflection on those choices, which is necessarily detrimental to ease of use.

Much tool criticism stems from the observation that tools are not neutral. This non-neutrality is the first of the so-called "Kranzberg's laws", stating that "technical developments frequently have environmental, social, and human consequences that go far beyond the immediate purposes of the technical devices and practices themselves" (Kranzberg, 1986: 545). It is worth remarking that Melvin Kranzberg was not really normative about neutrality. Tools are not neutral, Kranzberg says, simply because their effects are observably different from their purposes. Of the many different ways to make this argument, one in particular has been favored in the social science and humanities (SSH) literature on tool criticism: easy-to-use tools have detrimental effects because their users are let free to exploit them without understanding them (Tenen, 2016; van Es et al., 2021).

Much tool criticism also assumes that tools embody methods. Rieder and Röhle (2017) write that Gephi "performs a method" while for Tenen (2016), "the tool can only serve as a vehicle for methodology." Van Es et al. (2021) quote Dobson (2019: 8) to say that tools "require some formalization of the methodology". In this argument, the method is the purpose of the tool, and the primary concern is the circulation and influence of this method. Some tools threaten "the critical and interpretive traditions in the humanities and social sciences since the concepts and techniques embedded in these tools are often borrowed from the empirical sciences or corporate contexts" (van Es et al., 2021: 47; see also Drucker, 2014; Dobson 2019; Masson, 2017; Rieder & Röhle, 2017). This danger is addressed from multiple positions. One position addresses it from the outside: where does the tool come from? Who produces it? Who promotes it? With which resources and agendas? For instance, the tool may be "assigned such values as reliability and transparency" (van Es et al., 2021: 47). Another position addresses the danger from the inside: what is the tool's own responsibility in making certain worlds possible and visible while silencing others? In this paper, we focus specifically on this kind of internal responsibility of the tool as a material-semiotic entity.

In the SSH literature on tool criticism, tools are made materially responsible for a variety of problems, including being "vehicle[s] ... for [their] claimed self-evidence" (Ruppert & Scheel, 2019: 242), "pass[ing] as unquestioned representations of 'what is'" (Drucker, 2014: 125), making methods "accessible to broad audiences … without having acquired robust understanding of the concepts and techniques the software mobilizes" (Rieder & Röhle, 2017: 118), or concealing "built-in assumptions and biases" (van Es et al., 2021: 50). In short, tools fail to communicate some of their methodological tenets, and/or are too easy to use for their users' own good. This is widely identified as a blackboxing problem (Rieder & Röhle, 2017; van Geenen, 2020; Paßmann & Boersma, 2017): an "impossibility to know the method" and/or an "invisibilization of the mediation" (Jacomy and Jokubauskaitė, forthcoming).

Characteristic of this literature is the consensus that the tool is responsible for being a black box, regardless of some ambiguity about what it means. Some authors refer to what Jacomy and Jokubauskaite (forthcoming) call "blackboxing by embodiment," understood following Latour as "the way scientific and technical work is made invisible by its own success" (1999: 304). Other authors refer instead to what Jacomy and Jokubauskaite (forthcoming) call "blackboxing by inscrutability": the observation that some tools are "opaque decision systems" (Guidotti et al., 2018) that cannot be adequately subjected to methodological inquiry. Blackboxing by inscrutability is most prevalent in discussions about algorithm accountability and explainable AI (overview in Wieringa, 2020), and is probably what comes first to mind for the computer scientist who designs the algorithmic techniques embedded in digital tools. However, addressing this kind of blackboxing does not entail much more than exposing the method somewhere to enable scrutiny. Blackboxing by embodiment, on the contrary, involves the affordances of the tool. It has been largely discussed in the field of Human-Computer Interaction (HCI) (Dourish, 2004; Kaptelinin and Nardi, 2012) where CTP is also popular (Dourish et al., 2004). In HCI, like in tool criticism from the digital humanities (DH), blackboxing is implicitly a matter of embodiment and therefore intimately associated with ease of use.

For Tenen, "any 'ease of use' gained in simplifying the instrument comes at the expense of added and hidden complexity" (2016; see also van Es et al., 2021). This point is imported from HCI, where ease of use is part of the concept of usability, as defined by the norm ISO 9241 (Quesenbery 2001). Usability "is an essential concept in HCI, and is concerned with making systems easy to learn, easy to use, and with limiting error frequency and severity" (Issa & Isaias, 2015). Ease of use is one the goals that HCI aims to achieve; but HCI also acknowledges the tradeoff between ease of use and complexity as a design constraint. The tool maker's ideal "is to support different user levels of complexity, or to hide complex features, to allow both ease of use and powerful features … This works to some extent, but building complex systems involves complex design thinking. … There are no 'magic bullets' here … Power/Flexibility and Usability are usually at odds with each other, since simplicity tends to correlate with usability" (Murray, 2004: 11-12). In DH as well, ease of use is considered by some "one of the most desirable characteristics for any given tool" (Morgan, 2018: 212). Morgan thus criticizes Tenen's dichotomy: "Tenen characterizes a preference for easiness as a sort of intellectual laziness or lazy thinking, when more attention to method is warranted (118). In some cases, this critique is highly applicable; in others, it fails to take into account that the preference for easiness is influenced by a lack of infrastructure. … Out-of-the-box tools, which might be

better characterized as 'entry-level' DH tools, are arguably fulfilling a community need" (2018: 220). Scholars in digital humanities tend to debate whether the benefits of ease of use are greater than the problems it causes, whereas ease of use as a goal is a given in HCI; but both fields agree with the existence of a tradeoff between ease of use and exposed complexity. This point is thus recurrent in the literature on tool criticism. We will refer to it as the *black box tradeoff*.

The black box tradeoff states that the easier a tool is to use, the more it hides complexity and is blackboxed (see fig. 3). This point corresponds to Tenen's (2016) and is widely echoed (Dobson, 2019; Rieder & Röhle, 2017; van Es et al., 2021; van Geenen, 2018). Easy-to-use black boxes, so the argument goes, lead users to "lazy thinking" (Tenen, 2016) and adopting "unreflexive" practices (Dobson, 2019: 3). Implicitly, critical thinking lies on the other end of the tradeoff, where complexity is exposed at the cost of an increased difficulty of use. Note that exposing complexity, in this tradeoff, is a necessary but not sufficient condition for CTP, contrary to what fig. 3 may suggest. Also remark that the ease-of-use axis applies to the user and the tool at the same time.

```
      BLACK BOX                          CRITICAL THINKING



       <────────────────────────────────────────────────>
                        Ease-of-use axis

    The tool is                              The tool is
    EASY TO USE                          DIFFICULT TO USE
    HIDING COMPLEXITY                  EXPOSING COMPLEXITY


    The user is                              The user is
    LAZY THINKING                          HARD THINKING
    UNREFLECTIVE                              REFLECTIVE
```
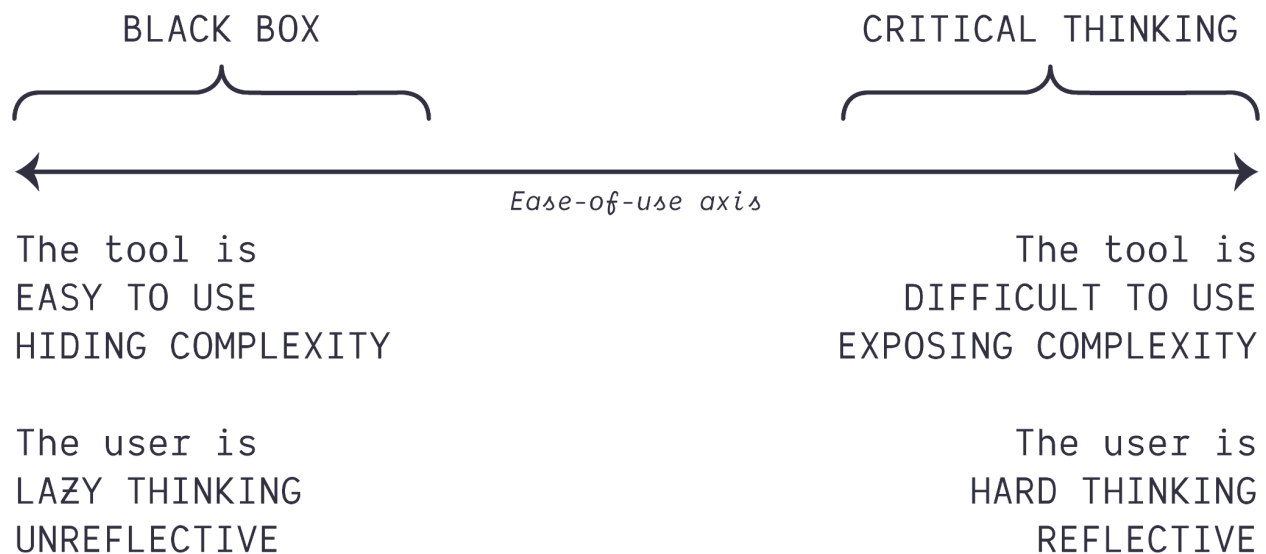
Figure 3: The black box tradeoff: one cannot make a tool that is easy to use while exposing complexity, which implies that the user cannot be critical of the tool if it is too blackboxed.

Also note that the HCI version of this tradeoff is different. It is not just the tool or the user, but the triple combination of a tool, a person, and a situation, that falls somewhere on the ease-of-use axis. HCI acknowledges that ease of use and blackboxing also depend on the user and the situation. The tool designer generally aims at addressing multiple needs at the same time. As Murray (2004) suggests, the designer can offer more complex features in places of the graphical interface that remain hidden from beginners. It is then possible that, for a given tool, different users in different situations fall in different places of the tradeoff. However, Murray also acknowledges that there is a limit to this, and that the tool will necessarily lean on one of the sides of the balance.

Last remark, in case it was not clear enough: we criticize the black box tradeoff for being too simple to explain what is at stake in tool making. Our motivation for expliciting it is to make our argument clearer, starting with highlighting the problems with this tradeoff.

The Mjölnir paradox (or the problem with the black box tradeoff model)

The black-box-tradeoff model implies that tools are either catering to beginners, who cannot be trusted or expected to have CTP, or to seasoned experts, from whom CTP can be expected and to whom tools with sufficient complexity must be designed. Unfortunately, this fails to address that beginners are just as much, if not more, in need of CTP as experienced users. It also fails to address that many experienced users are looking for ease of use and have, in fact, become experienced precisely in obtaining that despite the complexity of certain tools. Ease of use is not the prerogative of beginners, nor CTP that of advanced users.

In the black box tradeoff model, the aspiration of the user is somewhat irrelevant. Yet, as the HCI literature correctly identifies, usage is largely shaped by the situation (Murray, 2004). Seasoned users will occasionally seek one-click solutions with their tools, regardless of the level of complexity exposed by the design: time-constrained environments may not allow for much reflexiveness (in the moment); recurring monitoring checks call for ease of use; quick and dirty explorations can afford shortcuts. The black-box-tradeoff model does not help us understand why easy-to-use tools are (supposedly) dangerous to beginners who seek reflexiveness, nor why hard-to-use tools are (supposedly) safe to experts who seek ease of use. It just assumes that the design is the determinant factor and therefore suggests that easy-to-use tools make users think lazily, and/or that lazy users are drawn to easy-to-use tools. This leads to a surprisingly patronizing perspective: beginners are asked to stop using the easy tools that make them think lazily, and *somehow* learn to use hard tools where the complexity of the method is exposed, to become reflexive hard-thinkers. We do not deny that those who voice that criticism have their own idea about how that knowledge should be provided, but it is not through the tool. To make matters worse, the model is also patronizing to tool designers, who are told to stop designing blackboxed, easy-to-use tools, because those are dangerous to critical thinking. In this model, the tool is never in charge of cultivating critical thinking: it either prevents it or requires it.

The black-box-tradeoff model thus prevents us from thinking about a tool design that promotes CTP while catering to ease of use, despite the fact that such a tool might be exactly what could induce CTP in hostile places. Helping beginners to get reflexive is a major design goal for tool makers in science and education, yet it has no room in the black-box-tradeoff model. To make matters worse, seasoned users are assumed to always want to profit from exposed tool complexity for CTP, ignoring the fact that they are often seasoned in doing precisely the opposite, i.e. contributing to the embodied black box.

Seen from the point of view of the black box tradeoff model, then, the ideal CTP tool is somewhat akin to Mjölnir, Thor's mythical hammer, that can only be used by the worthy (fig. 4-a). Unfortunately, such a tool offers little help in terms of making the unworthy worthy. It would come with the weird requirement that users learn how to use it without actually using it. It also cannot exist, because users who are not interested in CTP worthiness can and will invent

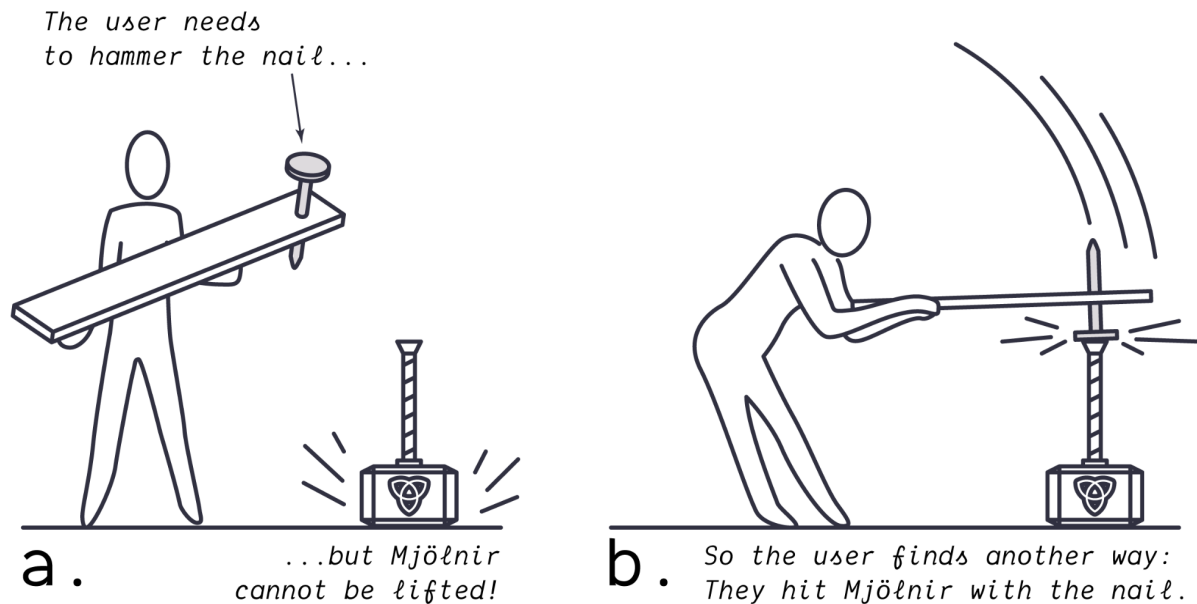unexpected ways to use the tool for meeting their needs anyway (fig. 4-b).

The user needs to hammer the nail...

...but Mjölnir cannot be lifted!

a.

So the user finds another way: They hit Mjölnir with the nail.

b.

Figure 4: The Mjönir paradox: the ideal tool for the black-box-tradeoff model is Mjölnir, Thor's hammer, which can only be used by those deemed worthy. In theory, the untrained user cannot use the tool (lift Mjölnir) which protects it against misuse (a). But in practice, the user can always find a way to meet their needs, which may result in even worse misuses (b).

Users cannot be prevented from learning, and they cannot be prevented from meeting their needs, even if it means misusing the tool and taking methodological shortcuts. Tool design can partially shape the user's behavior in many ways, but neither by forcing them into or preventing them from doing certain things. Incentivizing reflexivity and critical thinking through tool design is a good thing, we all agree on that. Most users, in most situations, are willing to learn to some extent, and the tool design should facilitate it. But how to deal with users who only want to reach their operational goals and do not want, or cannot afford, learning in the process? We cannot prevent them from repurposing the tool in utilitarian ways. With Gephisto, we explore an answer to that challenge. Gephisto looks like what such users want, because it does not require any background to be used; it's the lazy-thinking tool par excellence. Yet we argue that it contributes to the user's CTP. Before we explain this paradox, let us look into Gephi, a tool with pretty classic design intentions, as a point of comparison. As we will see, the practices around Gephi already hint at shortcomings of the black-box-tradeoff model.

Gephi, a counterexample to the black-box-tradeoff model

Gephi, a popular network analysis and visualization tool, was designed with the intent of avoiding blackboxing, in a perspective similar to the black-box-tradeoff model. The instrument's authors argue that "social scientists cannot use black boxes, because any processing has to be evaluated in the perspective of the methodology" (Jacomy et al., 2014: 2). They refer here to

blackboxing *by inscrutability* (inaccessibility of the method) as something to avoid. Their design intentionally exposes the complexity of the method to lead the user to engage with it. "The visualization of a network involves design choices. We think users have to be aware of the consequences of these choices. The strategy we adopt in Gephi is to allow users to see in real time the consequences of their choices, learning by trial and error. Interaction, we believe, is the key to understanding" (Jacomy et al., 2014: 11). Gephi's designers not only expose (some of) the complexity of the method, but also force the user to confront it.

As an emblematic example of this strategy, we focus on a specific design point: the choice of a layout algorithm. Visualizing a network requires picking a node-placement algorithm and its settings. Different algorithms exist, and they produce sensibly different results. Yet, no algorithm is inherently better than the other, each one has its strengths and weaknesses. Gephi's designers feared that offering a default algorithm would, in practice, make that choice and its arbitrariness invisible to many users. As a result, Gephi's design intentionally forces the user to face that choice: applying a layout requires picking an algorithm in a list, sorted by alphabetical order, where no item is particularly salient. There is no way around intentionally making that decision and in that sense, on this specific matter, the complexity of the method is exposed. However, as Jacomy and Jokubauskaitė (forthcoming) observe, the community of Gephi users has stabilized a layout choice by default as part of the "best practices" presented in tutorials or "speed courses", as a recipe shared between users. That algorithm is also presented as "default" in the abstract of the paper that describes the algorithm (Jacomy et al., 2014) and it has been developed by the same team that is also behind Gephi. Despite the intentional design of Gephi's user interface to avoid it, a default layout choice has *de facto* emerged from practices. Through the production of cultural artifacts, the user community compensated for the lack of a clear, simple path to applying a layout.

Other design points were similarly tamed: which settings to use, which filters to use, which statistical metrics to compute… As Jacomy and Jokubauskaitė (forthcoming) comment, "Gephi yields to utilitarian pressure in time-restricted situations where users are the most willing to trade methodological awareness for efficiency." Its design was not sufficient to prevent what Tenen (2016) calls "lazy thinking", because the user community overcame the resistance of the interface by producing its own shortcuts. Contrary to what the black-box-tradeoff model states, exposing the method did not lead to less ease of use and lazy thinking.

The case of Gephi highlights the importance of user aspirations. The user production of cultural artifacts could (and should) be considered a contribution to the tool itself, in a broad sense. Users often act as co-designers of the tool, and the tool's materiality is not bound to its source code and application, but extends to the cultural artifacts around it. This is especially crucial when we look at what some users perceive to be "*the* method", where its materiality lies, and who created it. Gephi certainly makes a number of methodological choices, such as which algorithms to implement or not, but it remains agnostic about which actions the user must take. As we have seen, the lack of guidance is an intentional design choice. Gephi's user interface does not push a *specific* method, it merely offers a methodological landscape, and prompts the user to explore it by trial and error. The "Gephi method" (as in e.g. Hemsley and Palmer, 2016) exists only as a recipe, or a set of recipes, documented in various cultural artifacts. In that case,

exposing the methodological complexity in no way prevents shortcut recipes to emerge as the *de facto* method, providing the missing path to ease of use and "lazy thinking" to which some users aspire to. It even suggests that rejecting ease of use might be counterproductive, when it incentivises users to devise their own shortcut recipes to obtain quickly the results they expect, in disregard of their own methodological awareness.

Beyond the black box tradeoff, users decide on their learning path

Instead of looking at the tool as a whole, we bring a bit of sophistication by looking at the decisions it requires. We now think of the tool as a series of methodological choices: what to do and in which order, which settings to use… This allows reformulating tool practice in terms of how the decisions are made. We focus on a single choice, for instance which layout algorithm to use in Gephi. Two things are worth remarking. First, the choice is not necessarily visible to the user. It might be concealed by the tool, or semi-hidden in an obscure submenu where some users will miss it. Second, the choice is not necessarily made by the user. It might be made by the tool itself, or by a combination of factors where the tool and the user play a role, for instance if the user keeps the default settings. The decision might involve external knowledge, for instance when the user replicates a tutorial. Making a decision is no simple matter. To keep things manageable, we propose a basic model of what it takes to build an informed decision.

The ladder of informed decision. We summarize the user's ability to make an informed decision as a ladder. The user needs three different things to make an informed decision, that we order as a series for simplicity (fig. 5). First, the user needs to be aware that the choice exists. Second, they need to know all of the options offered to them. Third, they need to know what each option entails. Then, and only then, can they assess which option is the best in a given situation. This is what we define here as an informed decision: assessing all of the options as a function of what they entail in the current situation, to pick the most appropriate.
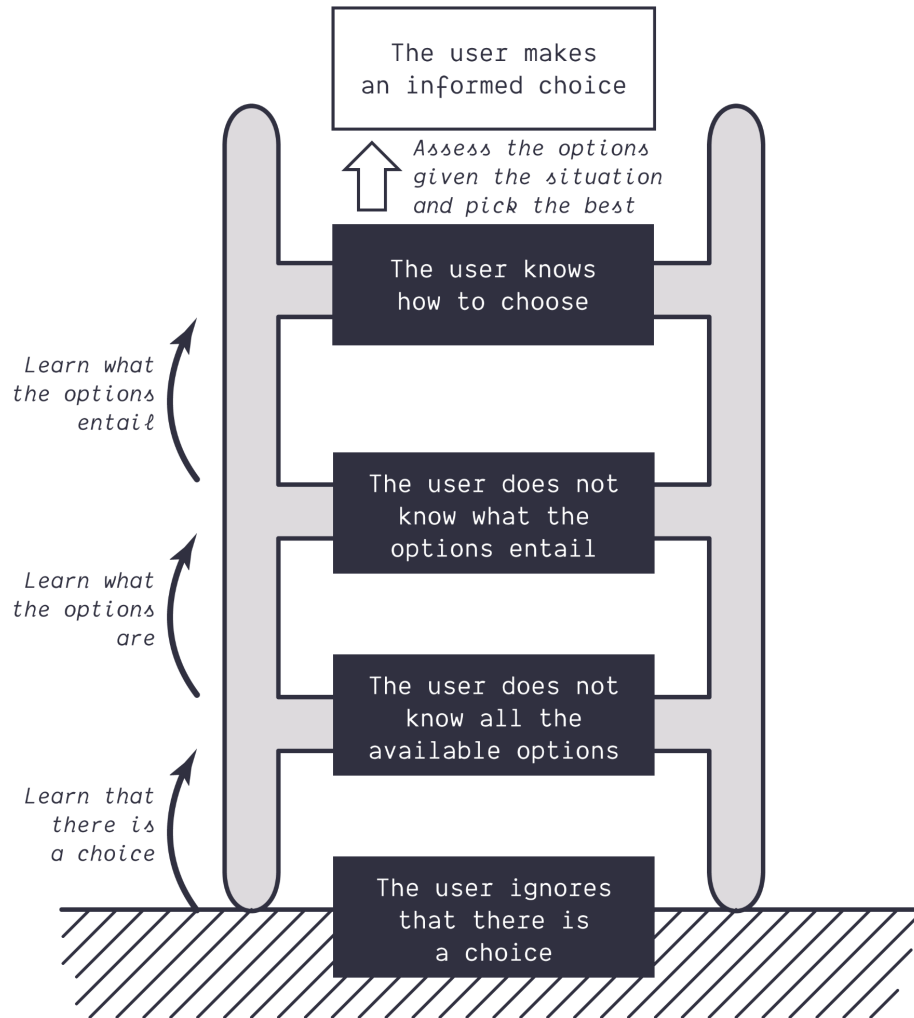
Figure 5: The ladder of informed decision. The user's ability to make an informed decision is built in three successive steps (on the left).

Making the decision corresponds to hard thinking: it requires comparing different options and assessing a specific situation; it takes effort. Climbing the ladder is also demanding, but in a different way: it only takes a one-time effort to get to the next step. The three steps are to know that there is a choice, to know what the different options are, and to know what they entail (fig. 5). That process is more gradual than the metaphor suggests. Learning what the different options entail will probably not happen at once, but through multiple moments of exploration. Climbing the ladder is demanding, but only requires a one-time effort, insofar as the user cannot easily be stripped away of their ability to choose, contrary to the decision making itself, that demands an effort each time because the situation varies. We assume here that the user will not unlearn what the options are, nor what they entail; although they might disregard what they know—we will return to that. Climbing the ladder of informed decision is *learning*, not hard thinking. Learning is a prerequisite for informed decision making, where hard thinking lies. In

this perspective, fully mastering a tool boils down to learning how to make informed decisions for all the choices it presents.

We call a *learner* a user who needs to use the tool, without fully mastering it, yet wants to learn something out of it. Every user is a learner to some extent, except maybe the most expert users. We use that term to discuss things that are specific to learning situations, and not only to beginners: with rich tools, one can know a lot and still learn. But it is worth reminding that beginners are learners too, and that their supposed laziness often arises from the learning situation. In other words, what applies to beginners also applies to more advanced users, to some extent, when they face their own limitations (with the tool).

The learner faces a dilemma regarding where to invest their time and energy. They must spend their mental resources on two different things: learning (how to make better informed decisions), and doing (reaching their operational goals). As Jacomy and Jokubauskaitė (forthcoming) narrate, users often come to a tool like Gephi with a need, an expectation, a result to obtain. Reaching this operational goal and learning how to use the tool compete for the learner's limited attention resources. The more time-restricted the situation, the greater the tension between the two. To facilitate our argument, we express that tension as a two-dimensional space (fig. 6), where the vertical direction represents learning, i.e. the progress of making informed decisions for all the choices presented by the tool, and the horizontal direction advancing towards operational goals. The key difference with the black-box-tradeoff model is to acknowledge that the user is free to navigate that space. The tool design may influence the user's path in that space, but it does not determine it. Different users, or the same users in different situations, may walk different paths. In a time-constrained situation, they may prioritize obtaining results over learning the tool, while they may not during a course. Note that the user may only move from left to right and bottom up, as we assume that they do not unlearn nor undo their progress towards their operational goals. In this space, going up represents learning how to make better informed decisions (fig. 7 on the left). Remember that we see the tool as a set of choices, each with their own ladder of informed decision to climb. Going to the right represents advancing towards operational goals, which we call "doing" for simplicity (fig. 7 in the middle). The limited resources of the user create a tradeoff between learning and doing (fig. 7 on the right). That tradeoff is a dilemma for the user because learning contributes indirectly to obtaining better results faster, but postpones obtaining them in the short term. Learning is an investment. Yet using the tool also contributes to learning, so even if learning is your priority, you should probably still use the tool. A pretty rational strategy, in such a situation, is to adopt a flexible approach where one learns when necessary and try obtaining results if one can. This ties back to the well-known usability problem that users only turn to documentation when they encounter a problem (Novick & Ward, 2006; Rettig, 1991).
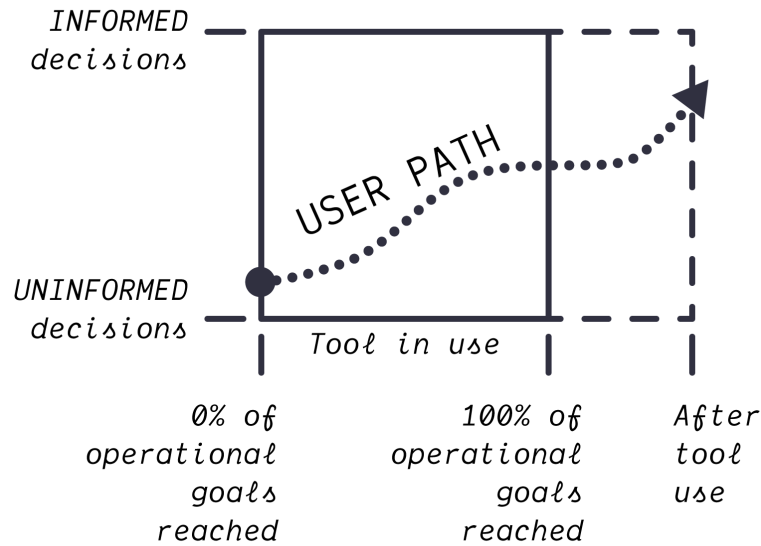
Figure 6: When using a tool, the user navigates a space where they learn how to make better decisions while reaching their operational goals. They are essentially free to take their own path, as the tool design can neither prevent them from learning nor from reaching their goals. Important remark: not all the learning happens while the tool is in use; it may happen before or after as well (dashed rectangle on the right; see also fig. 10).
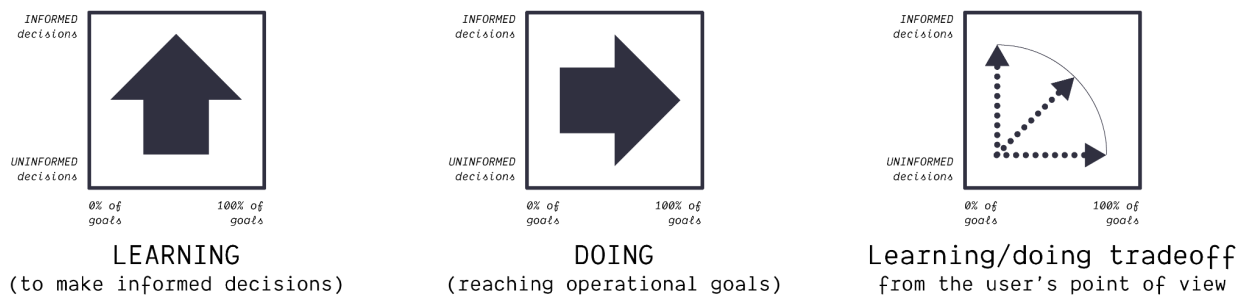
Figure 7: The user may learn how to make informed decisions (vertical axis) and/or use the tool to reach their operational goals, i.e. "doing" (horizontal axis). Their resources being limited, there is a tradeoff between learning and doing.

Translating the black-box-tradeoff strategy in this framework shows what is wrong with it. It tries to prevent unseasoned users from using the tool, forcing them to learn the tool before using it (fig. 8 on the left). Not only is this abdicating from fostering CTP through tool design, but it simply fails because the tool design cannot prevent users from finding or inventing shortcuts to meet their operational goals (fig. 8 in the middle). It is the user who decides the path. We cannot block certain areas: the user can always learn and/or advance their operational goals if they want to. The tool cannot *oppose* the user. But the tool may *influence* them in various ways. What we seek, as tool designers, is to incentivize the user to learn (fig. 8 on the right).

Figure 8: Tool makers may be tempted to force the user to learn the method before using the tool (left), but this does not work as the user can always find or invent shortcuts to prioritize doing over learning (middle). A better strategy is to incentivize the user to learn (right).

What is it to be uncritical in this context? When a learner is "lazy thinking", to reuse Tenen's word (2016), that is often because they had to compromise between learning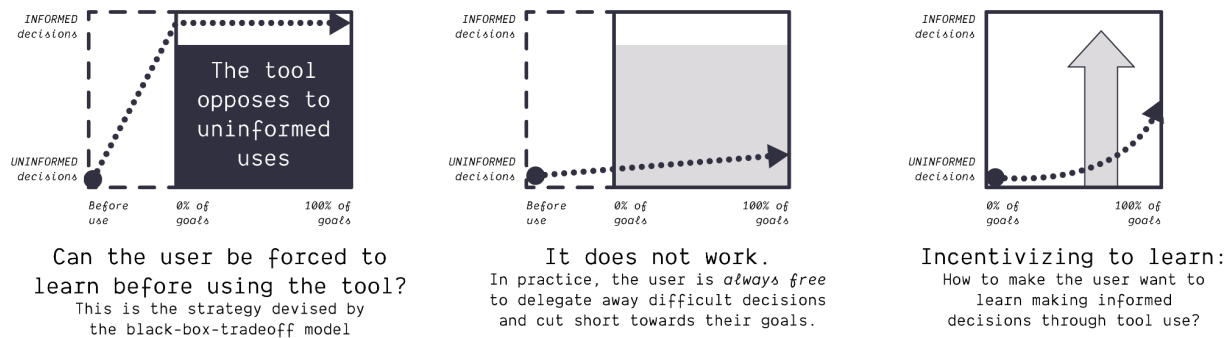 and reaching an operational goal. Not all the choices presented by the tool could be mastered, and some decisions had to be taken in uninformed ways so that results could be obtained. The learner had to resort on default settings, external recipes (e.g., replicating a tutorial), and/or some amount of arbitrariness. Learners, by definition, lack options. Calling this "laziness" wrongly assumes that learning is a mandatory duty that everyone can afford in every situation. In the black-box-tradeoff model, critical thinking is the prerogative of hard-to-use tools, inaccessible to the uncritical beginners (Mjölnirs, see fig. 4). On the contrary, the learning/doing space helps understanding critical thinking as the result of an intentional user behavior, only partially influenced by the tool design. CTP is never granted, as the user has to build it. And symmetrically, CTP is never prevented against the user's will. No tool is that good that it makes all users critical, but no tool is that bad that it prevents all users from being critical. More importantly, we should not confuse the inevitable learner misuses for laziness or for a lack of aspiration to reflexiveness. Learning entails some amount of uninformed decisions, and that should not be considered uncritical usage. But uncritical usage exists, and it is rather about abusively delegating one's decisions.

All users, not only beginners, may delegate their decisions. We call delegation the user's act of using a device or a knowledge support to pick a choice in their stead. Experts may for instance delegate their decision in the early stages of an exploration loop: they may start with default settings, assess the situation, then refine the settings to get better information. Here, the decision has been *delegated* to the tool through the default settings it offers. The user may also delegate to external artifacts like tutorials, papers, or other recipes. When a learner lacks the ability to make an informed decision, their *only* way to obtain results is to delegate their decision (fig. 9). In that sense, delegating can be considered an informed act for the expert, and not for the learner. The expert has already climbed the ladder of informed decision. The learner, however, may normalize the reliance on delegation as the go-to use of the tool, never building the ability to make an informed decision. This is *actual* uncritical usage, and an inherent problem with delegation. Yet delegation also has benefits, as it allows the learner to obtain results, which enables learning by experience (if the user wants it). Delegation has both

advantages and drawbacks. It is dangerous to learners, and at the same time, productive to experts.
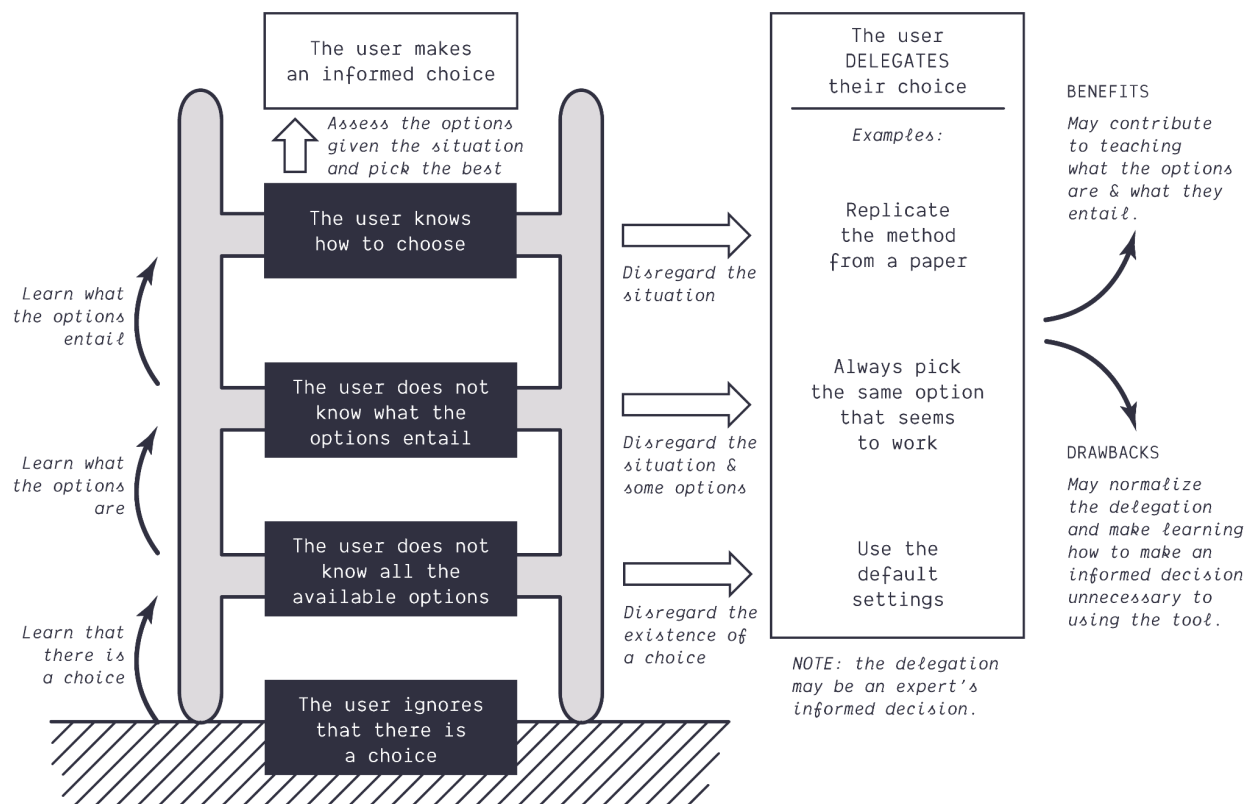
Figure 9: The user may delegate their decision to the tool itself, or to external resources like tutorials. The delegation has both benefits and drawbacks: it helps obtaining results, which may contribute to teaching how the tool works, but it may also normalize the delegation as a methodological shortcut.

Delegation is related to what the black-box-tradeoff model calls ease of use. Let us enumerate the different ways the tool's materiality may contribute to shape the user's ability to make informed decisions. (1) The tool's power is **limited**: it may neither impeach nor force the user to learn. Furthermore, it may not prevent the users from delegating. As we have seen with Gephi, users may invent and disseminate their own shortcut recipes. (2) The design may however incentivize learning by **exposing complexity**: showing that a choice exists; what its options are; and offering feedback about what they entail. The design may conversely disincentivize learning by obfuscating the method (blackboxing by inscrutability). (3) The design may enable or incentivize **delegation**. This last item is specifically the problematic ease of use of the black-box-tradeoff model: commodity features that alleviate the cost of decision making. These features may be default settings, guidance features like advice or embedded tutorials, simplified choices, and more generally every affordance aimed at facilitating decision making. The question is then: considering that delegation is inevitable, is there a way to make it more

productive while mitigating the threat it poses to climbing the ladder of informed decision making? Fighting delegation is a losing battle, but we may make it more productive to CTP.

Not fighting delegation is the fundamental premise of Gephisto's design; we call it the *Faustian bargain strategy* for reasons we will soon expose. To understand how this could be productive to CTP, we must take a step back and look at how learners do learn over time. After producing a network map with Gephisto, they will have opportunities to reflect on it. For instance, they may discuss it with colleagues, or share it and get feedback, etc. These reflexive moments may teach valuable lessons on what different choices entail. Learners do not only learn when the tool is in use, but also when the tool is not in use (fig. 10). Different scenarios give birth to different paths. Let us see some examples. The most uncritical user, the user who never learns, has a completely flat path (fig. 11 on the left). The expert user does not learn either, but that is because they have already mastered the tool (fig. 11 in the middle). A steady learner will gradually master all the choices, reaching the top as quickly as they prioritize learning over doing (fig. 11 on the right). Aside from the learning pace, the learning may happen during the reflexive moments (fig. 12 on the left) or on the contrary during use (fig. 12 on the right). We often assume that the users will learn from using the tool: this works well with Gephi, because its design exposes many choices while it disincentives certain delegations, for instance by not offering a default layout algorithm. But this strategy fails to contain delegation-seeking users, who will rely on recipes to cut short any exposed complexity. The Faustian bargain strategy is aimed at these users specifically.
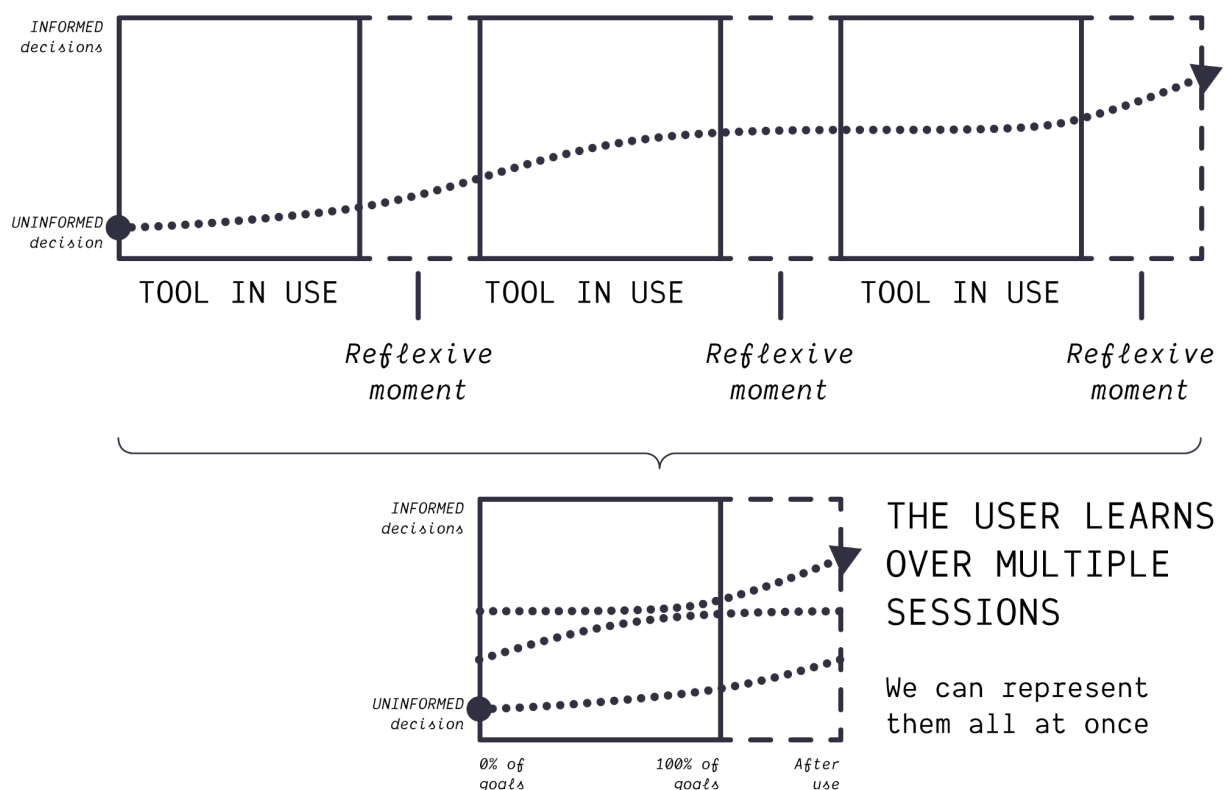
Figure 10: we assume that users want to reach their goal and improve altogether, even when they prioritize doing over learning. We also assume that this happens over multiple occasions of using the tool.
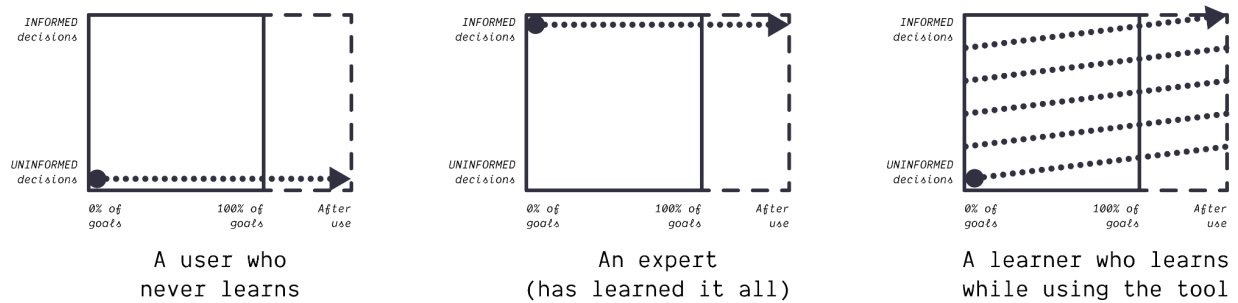
Figure 11: Examples of different user paths.

Learning exlusively
when the tool is not in use

Learning exclusively
by doing
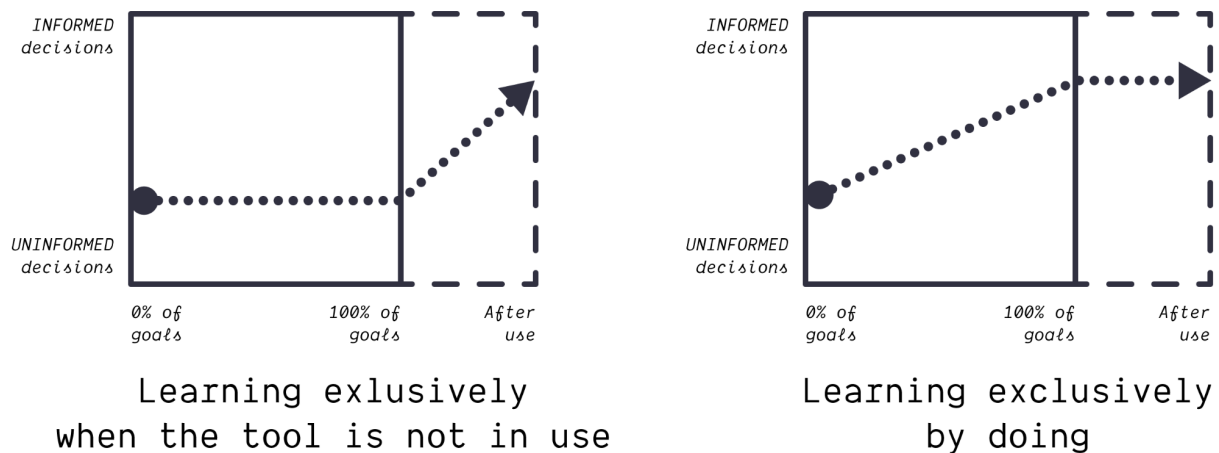
Figure 12: Depending on the situation, learning may happen when the tool is in use (right), and/or not in use (left). When the tool is in use, the user needs to make it work and is more prone to prioritize doing over learning. When it is not in use, the user may have more resources to understand how it works.

The Faustian bargain strategy relies on a key property of the technical mediation: "reversible blackboxing" (Latour, 1994). Latour writes: "Why is it so difficult to measure, with any precision, the mediating role of techniques? Because the action that we are trying to measure is subject to 'blackboxing', a process that makes the joint production of actors and artifacts entirely opaque. … Open the black boxes; examine the assemblies inside. Each of the parts inside the black box is a black box full of parts. If any part were to break, how many humans would immediately materialize around each?" (1994:36-37) Here, Latour highlights that when a device breaks, it has to get unblackboxed to get repaired. It can then get blackboxed again to be put back to use.

Tools are typically blackboxed when in use, but not necessarily otherwise. The archetypal example comes from Merleau-Ponty's (1962) phenomenology of perception: the blind man's stick. When in use, it is *embodied*. The blind man ceases to feel it in their hand. It becomes part of their body, and they feel *through* it. A tool like Gephi is similarly blackboxed when in use, but may get unblackboxed in a different context. Jacomy and Jokubauskaitė (forthcoming) analyze that Gephi's blackboxing is precisely *by embodiment*. Delegation-seeking users will not open the black box of their tool when it is in use, but they may be incited to do so subsequently, when the tool is not in use anymore (fig. 13). This is why we devise the strategy of offering a form of delegation that can feed into the subsequent reflexive moment as a teaching moment. Gephisto is a "What if?" version of Gephi entirely based on this strategy. It delegates maximally by taking all the decisions, yet it tries to teach a lesson.
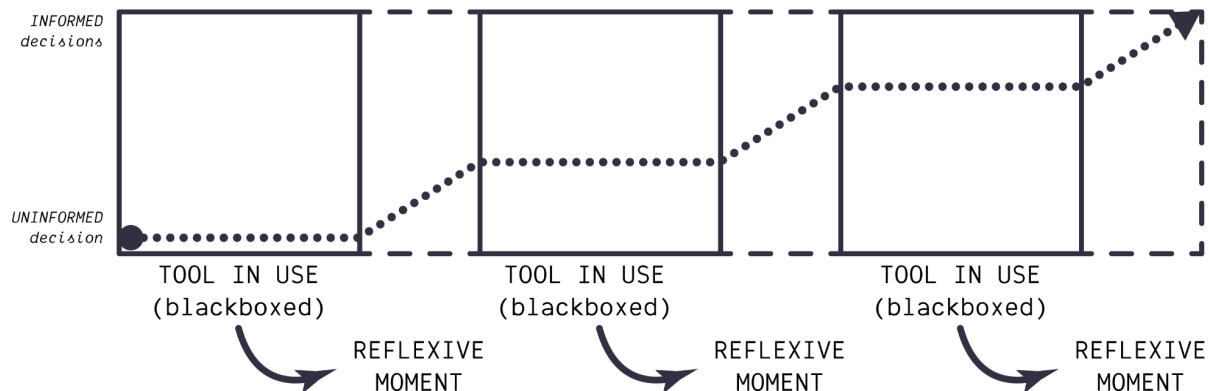
The Faustian bargain strategy:
Leveraging ease of use to feed reflexive moments.

Figure 13: The Faustian bargain strategy consists of supporting the delegation seeked by some users, but in a way that feeds reflexive moments where learning has a better chance to happen.

The Faustian bargain strategy

The Faustian bargain strategy is counterintuitive if you conceive ease of use as an alleviation of the burden of thinking, a common idea defended for instance by the HCI bestseller *Don't Make Me Think!* (Krug, 2000). The design challenge is paradoxical because it requires the coexistence of ease of use and hard thinking; which is also why it does not fit in the black-box-tradeoff model. But as reversible blackboxing reminds us (Latour, 1994), ease of use is not an obstacle anymore when the tool is not in use. To operationalize this idea, we draw inspiration from Mephistopheles, a demon with whom the protagonist Faust made a bargain, in a popular German tale. The tale loosely follows the trope of the evil djinn/genie, as featured for instance in Sapkowski's (2008) famous short story *The Last Wish*. The central element of the trope is a wish, object of a transaction between two agents: the demander and the granter. In the version of the trope that interests us, the granter teaches a lesson to the demander (e.g., "be careful what you wish for"). Indeed, the wish is the cliché plot device to deliver moral messages about the dangers of desire. But the content of the lesson is less relevant to us than the way it is delivered: indirectly. In this trope, the demander is betrayed either because the wish is implemented in an unexpected way, or because it has unforeseen consequences. In the end, the granter always wins (Mephistopheles gets Faust's soul; in the popular tale at least, not in Goethe's version). The mode of action of the wish granter is a good inspiration to design a tool, and technology in general, because it acts without dictating or requiring. Its influence is not a frontal force of resistance, but a lateral wind that deviates courses of actions without interrupting them. Technical objects precisely have the property to "act, displace goals, and contribute to their redefinition" (Latour, 1994). If we see the granter as the tool and the demander as the user, then the wish is the user's desire to obtain a result without making decisions. We already know that such delegation is problematic, so like the granter in the tale, we know that there is something wrong with the wish itself. We can therefore grant the wish and let the consequences teach a lesson. This means helping the user delegate as much as they want, but have them face the consequences. This way, ease of use can become a vessel for CTP… assuming that the user *actually* confronts the consequences. That is the difficult part.

We know that the user may ignore lessons anyway, so the Faustian bargain strategy is a bet. We bet on the following assumptions. First, we assume that the user wants to do something with their result. They will at least look at it and form an idea on whether or not it works for what they have in mind. We will leverage their potential frustration or desire to improve the result as a way to build a reflexive dialogue with the tool. Second, and most importantly, we assume that the choices embedded into the tool actually matter; and we mean it in a strong sense. Of course, we mean that different decisions lead to different results that, in turn, lead to different interpretations. But we also mean that these different interpretations are equally valid. In other terms, we assume that a number of methodological choices are fundamentally open. This is not a given. Indeed, objectively bad decisions do exist, like coloring nodes and edges the same as the background, which produces an illegible visualization. We discount objectively worse options, at least when we can reasonably detect them. A truly open choice happens when multiple non-obviously worse options remain. This openness may happen for many reasons: lack of consensus on what the better option is; consensus that we currently ignore what the

better decision is; the better choice may depend on the situation in a way that is problematic to assess… There is an endless list of reasons for the open-endedness of a methodology, and we will provide some examples that arose when implementing Gephisto. The important point is that delegating decisions is problematic because of their openness. The lesson of the Faustian bargain comes from revealing what is wrong with the wish; the wrongness, with the user's wish for easiness, is to deny that fundamental openness, to deny that valid results are a landscape to explore, and not a single destination.

Let us insist one last time that there is nothing inherently wrong with delegating a decision. We constantly delegate decisions because we think that a tool can make it better than us. That is precisely why tools, and technology, exist in the first place: to perform something for us. Any tool implies a delegation. This is why Gephisto cannot afford to be a bad Gephi, only delivering the empty lesson that tools may make bad decisions; this is completely obvious. Gephisto has to work well, it has to deliver results at least as good as what the user could do with Gephi. For instance, Gephisto's network maps have to be at least as readable as following a Gephi tutorial. The lesson taught by the Faustian bargain strategy is not that choices have consequences, it is that making decisions entails roaming a landscape of valid outcomes. That is why it matters for scientific instruments: methodology is a space with many roads, not a destination to reach.

The Faustian bargain strategy consists of never taking the same methodological routes. The tool takes certain paths, because it makes the decisions. But it never commits to the same ones, because that would invisibilize the existence of alternate paths. The results are therefore inconsistent. This inconsistency is uneasy for the user, even though the tool is easy when it is in use. The wish is fulfilled, because the user's need is met without making any decisions; but as the user realizes, the way this need is met is fundamentally unpredictable. The lesson is delivered when the user takes interest in the result they obtained. This is why the tool design must provide feedback that it is perceived as intentional, not as a glitch, and must expose which decisions have been made. The next section will provide examples. Let us wrap up the Mephistopheles metaphor: the expected user to come to Gephisto comes with the wish of a one-click Gephi, a tool that makes a network map without requiring them to make any decisions. Gephisto grants the wish but betrays the user on a technicality: the network maps produced are unpredictable. They are all equally good (in theory at least), but vary because the choices have multiple valid options. This reveals the implicit expectation that there would be a single path to a good network map. The user is taught the lesson that this is not the case: there are multiple valid paths. If the user wants to take one path in particular, they have to make decisions. It should depend on their situation, like their research questions, something that the tool cannot guess. If they decide to switch to Gephi and try making their own choices, then Gephisto has metaphorically owned their soul: it will have turned them to CTP. And we have done so through absolute ease of use, albeit in conjunction with exposing the complexity of the tool.

As a final remark to this section, note that there are multiple ways to build on the idea of the Faustian bargain. A possibility that was out of our reach but worth mentioning, was to offer a continuous path from delegation to informed decision making. Gephisto basically defaults on Gephi to offer a more critical practice, but a more ambitious take would be to have it all in a single tool. Assuming that the user decides that learning how to make better informed decisions

is worth their time, then it would be useful to build on top of the feedback that convinced them to start opening the black box, and give them a possibility to start tuning their decisions. THat is a design to investigate in the future.

Gephisto's design

Gephisto is a one-click tool to visualize a network. It inputs a network file (in GEXF for the moment) and outputs an image intended to be printed on a A4 paper sheet in good quality (a 20cm by 20cm square in 300dpi). It is implemented as a web application run entirely client-side (no data is uploaded on a server), and it handles networks of a few thousands nodes or less on an average consumer computer. Gephisto is an open source prototype (licensed under GPL v3.0), and not a commercial product. Gephisto is a methodological statement, the material incarnation of this article, making by different means the same points about ease of use and CTP; it is a research artifact. As Rieder et al. (2021) put it, "the tool *develops the user*, by presenting particular approaches to research they may not yet be familiar with. Toolmakers can calibrate this relationship to a degree: they may present the tool as a *statement* - 'this is how we think research should be done' - or as part of an exchange that feeds directly back into the tool itself." Gephisto is a statement about opportunities to develop the user.

Gephisto can be tested at this URL: https://bit.ly/gephisto

A notable design aspect is how apparent Gephisto's unpredictability is. Should Gephisto's trickiness be explicit or not? As the user can always decide on their path, we considered that it was more productive to have our intentions revealed, to build some complicity with the user thanks to the playfulness of the tool. That is why Gephisto's design only *pretends* to be deceptive, while actually revealing its intentions. We explicitly highlight that the results are different each time, and we dramatize it as a bargain with the figure of Gephisto in the graphical interface (fig. 14, see also fig. 1). This way, we ensure that the user is aware of the inconsistency of the produced network map, which is instrumental to the effect that we hope to produce.
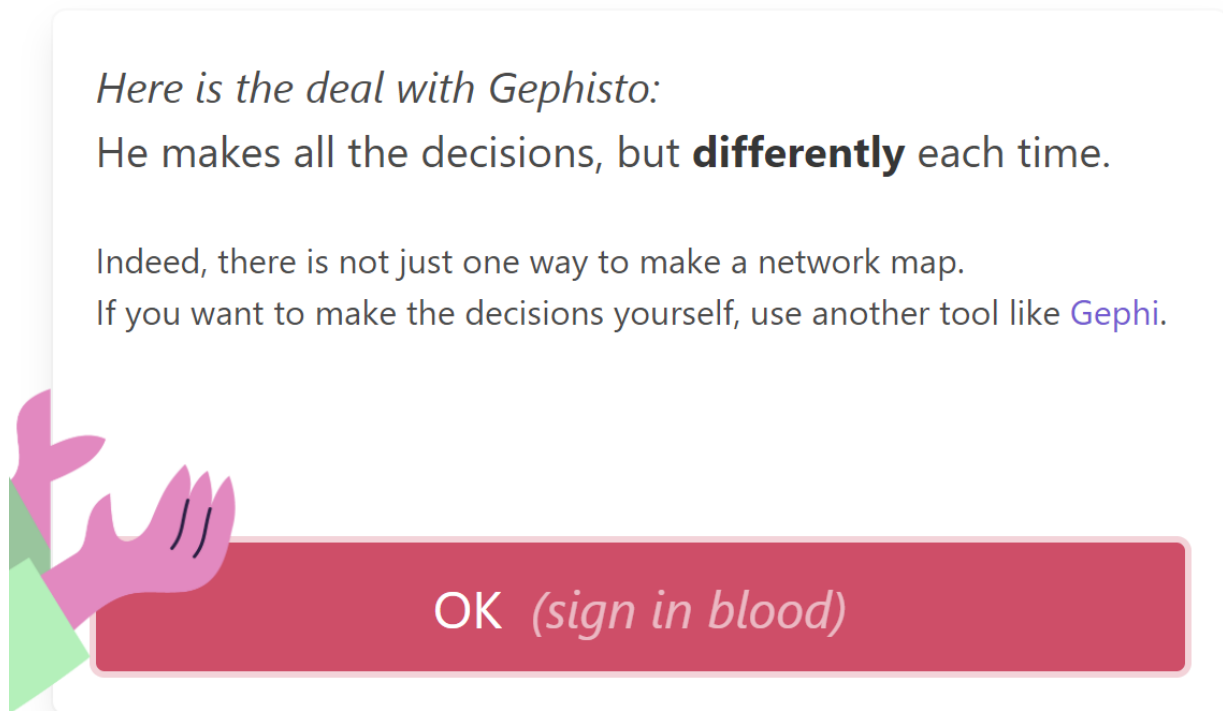
Figure 14: the message explaining Gephisto's principle. The drawback is also dramatized as a devil's pact involving a sign in blood. A bloody fingerprint appears when the user clicks on the button.

Variations of the results can be seen in figure 2, but it is worth highlighting worst case scenarios. In the network map presented in fig. 15, Gephisto made the wrong call, albeit for a good reason. It was confronted with a node attribute with just a few modalities: 0, 1, 2, 3 and 4. This could be two different things: an ordinal attribute, like the degree (number of links), or a categorical attribute. Ordinal attributes are suitable to node sizes, and categorical attributes to node colors. Unfortunately, integers can be both, and Gephisto cannot know for sure which one it is. When two options are valid, Gephisto picks at random. In this instance, the attribute corresponded to communities (modularity classes) and Gephisto wrongly assumed that it was an ordinal attribute, resulting in a visualization where node sizes are irrelevant (fig. 15). What the image displays is specified in the legend (fig. 15 on the right), but the image itself does not allow a very useful interpretation. Gephisto simply lacked the context to make a better decision, so it relied on randomness instead. In other instances, it made the right call and used that attribute for node colors (fig. 16).

The node coordinates are computed using the layout algorithm Lin Log. The gravity was set to 5.39e-5. For the layout, 5078 iterations were computed, in two batches with different settings, to optimize the rendering. The 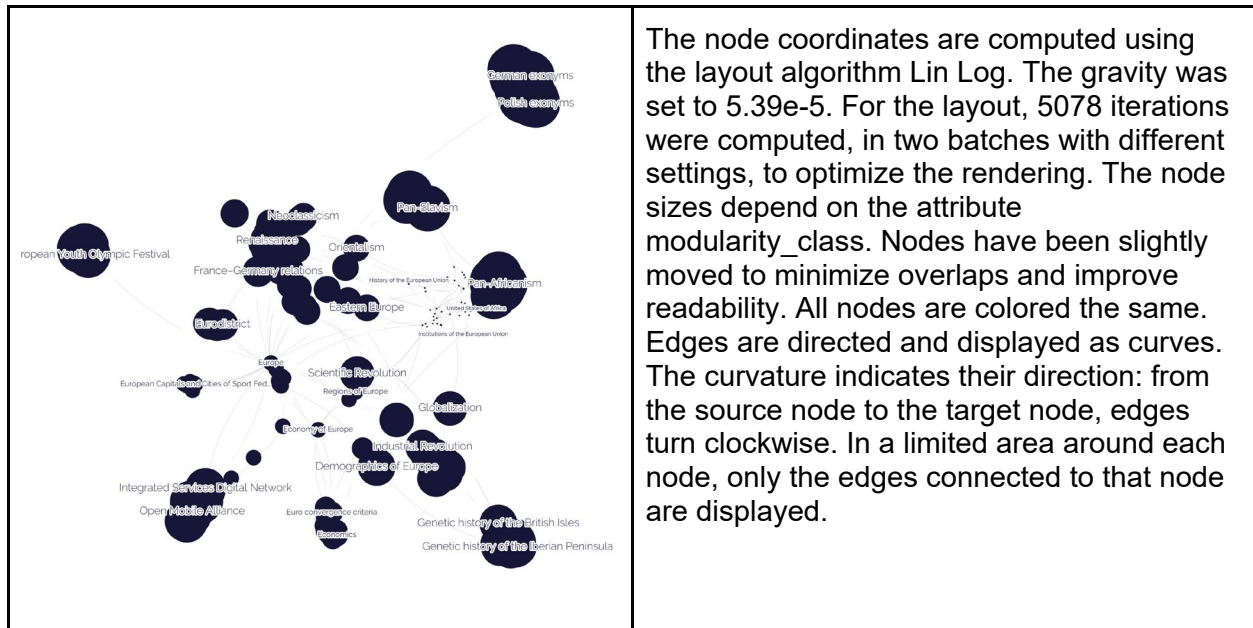node sizes depend on the attribute modularity_class. Nodes have been slightly moved to minimize overlaps and improve readability. All nodes are colored the same. Edges are directed and displayed as curves. The curvature indicates their direction: from the source node to the target node, edges turn clockwise. In a limited area around each node, only the edges connected to that node are displayed.

Figure 15: an example where Gephisto made wrong decisions. On the left, the network map produced. On the right, the corresponding legend. Gephis incorrectly assumed that modularity_class was an ordinal attribute, while it was categorical (integer modalities make both cases possible).

The node coordinates are computed using the layout algorithm Force Atlas 2. The gravity was set to 2.72e-2. For the layout, 1802 iterations were computed, in three batches with different settings, to optimize the rendering. The node sizes depend on the attribute degree. Nodes have been slightly moved to minimize overlaps and improve readability. The node colors depend on the attribute modularity_class: in blue is 3; in pink is 0; in green is 10; in red is 5; in yellow is 1; and in grey are the other modalities. Gatherings of nodes with the same modularity_class were highlighted by a color contour corresponding to their modality. Edges are directed and displayed as curves. The curvature indicates their direction: from the source node to the target node, edges turn clockwise. In a limited area around each node, only the edges connected to that node are displayed.
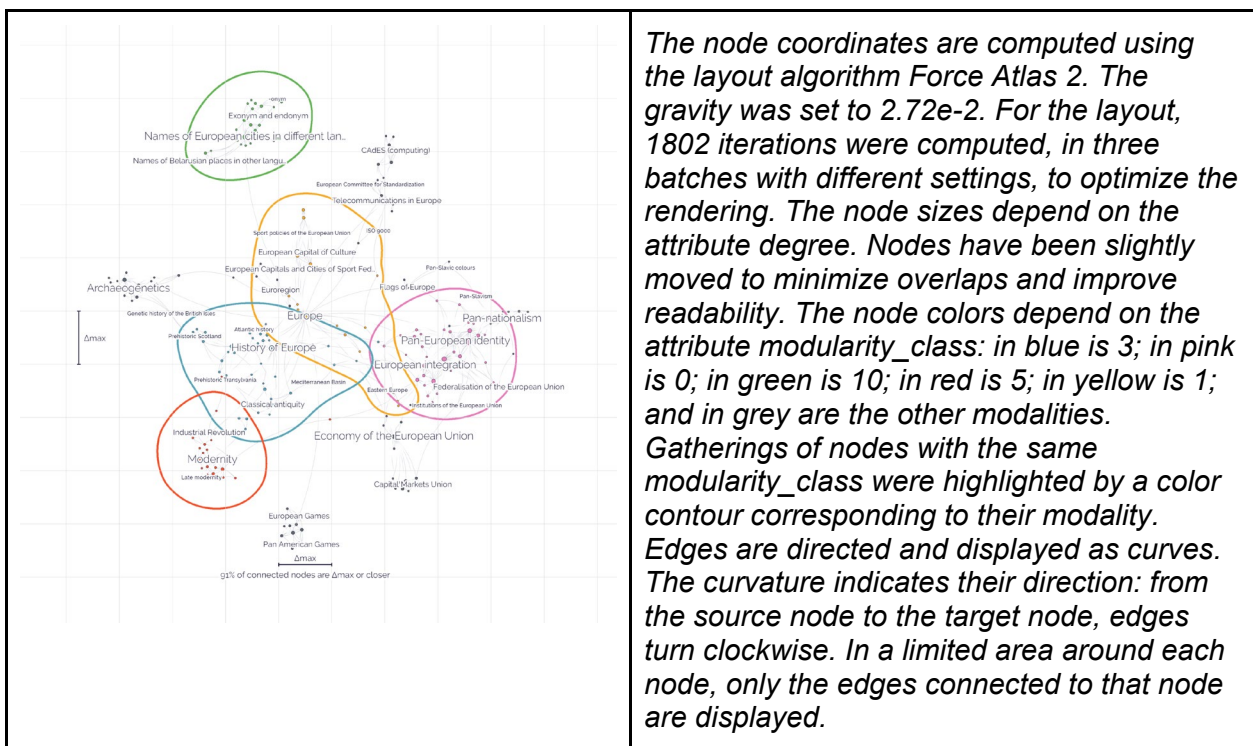
Figure 16: an example where Gephisto made the better decision with the same attribute (modularity class). It was used to color the nodes, because it is categorical.

Finally, here is a simplified version of the decision tree used by Gephisto to build a map. It involves a number of heuristics that we do not describe here, but they can be found in the commented source code[1]. This tree is a statement about what the making of a network map entails. We did not randomize all possible decisions, only those that had multiple valid options. For instance, we do not use a random layout because we consider it always worse than a force-directed layout. In that sense, the tree is also a statement of which choices, in the map making process, depend on the situation. Those are decisions that should involve the user. At the moment of writing, Gephisto makes exactly 19 decision points, highlighted below. Their probabilities have been tuned: for instance, the probability to display edges is high (85%), because it is generally, but not always, the better call. We featured below the probabilities for the 12 boolean choices (yes/no).

- Do we **draw the grid**? (50%) It provides context on the layout (Jacomy, forthcoming) but makes the image busier.
- Do we use a **dark mode**? (33%) Black on white is more suitable to print and white on black to display.
- The input network may already contain visualizing elements, like node positions or colors. Should we **reuse node sizes and positions** (25%), if they have any?
    - If not, do we at least **reuse the positions** (50%)?
        - If not, we must **pick a layout** (LinLog or Force Atlas 2), choose if we **favor quality over a quick result** (number of iterations), pick a **gravity setting**.
    - If not, do we use **different node sizes** (85%)?
        - If yes, we **pick a node attribute** (ordinal), an **average node** size and a **node size variance**.
        - If not, we only pick a single size for all nodes.
- Do we **use original node colors** (25%), if any?
    - If not, do we want **different node colors** (85%)?
        - If we do, we **pick a node attribute** (categorical) for node colors.
- Do we **display clusters** (33%)?
    - If yes, do we already have colors for nodes?
        - If we do, we use the same attribute to avoid confusion
        - If we do not, we pick a node attribute (categorical)
    - If yes, do we **display clusters as fills** (50%) or as contours?
        - If we display them as fills and nodes are not colored, we do not display node labels, to put emphasis on clusters. Useful for big networks.
- Do we **display edges** (85%)? It is sometimes productive to hide them.
    - If yes, do we **show edge weights as thickness** (75%)?

---

[1] https://github.com/jacomyma/gephisto

- ○ If yes, do we use **high quality edge rendering** (33%)? It is time consuming but makes the map more readable.

Conclusion

John Law (2009) has highlighted the difficulties of critiquing systems that perform without prescribing. He writes: "to be sure, systems of due process, including those that are successful, have effects. They are performative. But they are also messier than the descriptions or instruction manuals that accompany them. Rules, as Wittgenstein taught us, do not determine their application." Neither the design intent of a tool nor the methodology asserted by its documentation suffice to determine how users will use it in practice. Leading users to adopt critical tool practices takes more than normative statements because as we have contented earlier, they have an active role in deciding the tradeoff between learning and doing, between investigation how to make better informed decisions and pursuing immediate utilitarian goals such as obtaining a network map regardless of how it was built. One cannot enforce critical tool practice by rules imposed to the user.

Following Law's suggestion, we propose an "interference," a different form of intervention that does not aim at obtaining precise outcomes; an experimentation. We propose Gephisto, an alternative version of Gephi that does not require the user to take *any* decisions: it visualizes networks in one click. It is not a replacement, but a complement to Gephi, aimed specifically at users who cannot afford or do not want to reflect on what building a network map entails. We hope that Gephisto's tricky design, based on rendering unpredictable visualization due to the randomization of undecidable choices, will interfere with the user's project of delegating as much decision to the tool, and incentivize them to realize the existence of implicit decisions that shape the network map and its interpretation. In that sense, Gephisto explores a design perspective that could help beginners build critical thinking about their instrumented practices. We hope that understanding how users interact with Gephisto will help us build better scientific instruments in the future, and improve existing instruments like Gephi.

References

Dobson JE (2019). *Critical digital humanities: the search for a methodology.* University of Illinois Press.

Dourish P (2004) *Where the action is: the foundations of embodied interaction.* MIT press.

Dourish P, Finlay J, Sengers P and Wright P (2004) Reflective HCI: Towards a critical technical practice, *CHI'04 extended abstracts on Human factors in computing systems* (pp. 1727-1728).

Drucker J (2014). *Graphesis: visual forms of knowledge production.* Cambridge, MA: Harvard University Press.

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F and Pedreschi D (2018) A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5), pp.1-42.

Hemsley B and Palmer SR (2016) Two Studies on Twitter Networks and Tweet Content in Relation to Amyotrophic Lateral Sclerosis (ALS): Conversation, Information, and 'Diary of a Daily Life'. In *HIC* (pp. 41-47).

Issa T and Isaias P (2015) Usability and Human Computer Interaction (HCI). In *Sustainable Design* (pp. 19–36). London: Springer. https://doi.org/10.1007/978-1-4471-6753-2_2

Jacomy M, Venturini T, Heymann S and Bastian M (2014) ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PloS one,* 9(6), p.e98679.

Jacomy M (forthcoming) Connected-closeness: A Visual Quantification of Distances in Network Layouts, *Journal of Graph Algorithms and Applications*

Jacomy M and Jokubauskaitė E (forthcoming) *Unblackboxing Gephi: How user cultures shape their scientific instruments*

Kaptelinin V and Nardi B (2012) Activity theory in HCI: Fundamentals and reflections. *Synthesis Lectures Human-Centered Informatics*, 5(1), pp.1-105.

Krug S (2000) *Don't make me think!: a common sense approach to Web usability*. Pearson Education India.

Latour B (1994) On technical mediation - philosophy, sociology, genealogy. *Common Knowledge*, Fall V3(2), 29–64.

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

Law J (2009) The Greer-Bush Test: on Politics in STS, version of 23rd December 2009, available at http://www.heterogeneities.net/publications/Law2009TheGreerBushTest.pdf, (downloaded on 23rd December, 2009).

Masson, Eef (2017) "Humanistic Data Research. An Encounter between Epistemic Traditions." In: Schäfer MT and Van Es K (eds) *The Datafied Society. Studying Culture through Data*, 25–38. Amsterdam: Amsterdam University Press.

Merleau-Ponty M (1962) [1945] *Phenomenology of Perception*. transl. Colin Smith (NewYork: Humanities Press).

Morgan PC (2018) The consequences of framing digital humanities tools as easy to use. *College and Undergraduate Libraries*, 25(3), 211–231. https://doi.org/10.1080/10691316.2018.1480440

Munk AK, Madsen AK and Jacomy M (2019) Thinking Through The Databody: Sprints as Experimental Situations. In *Designs for Experimentation and Inquiry* (pp. 110-128). Routledge.

Murray T (2004) Design Tradeoffs in Usability and Power for Advanced Educational Software Authoring Tools. *Educational Technology*, 44(5), pp.10-16.

Novick DG and Ward K (2006) Why don't people read the manual? *ACM SIGDOC 2006 - Proceedings of the 24th ACM International Conference on Design of Communication*, 11–18. https://doi.org/10.1145/1166324.1166329

Paßmann J and Boersma A (2017) Unknowing Algorithms: On Transparency of Unopenable Black Boxes, In: Schäfer MT and Van Es K (eds) *The Datafied Society. Studying Culture through Data.* Amsterdam: Amsterdam University Press. https://doi.org/10.5117/9789462981362

Quesenbery W (2001) What does usability mean: Looking beyond "ease of use." *Proceedings/STC, Society for Technical Communication Annual Conference*, 432–436. http://www.wqusability.com/articles/more-than-ease-of-use.html

Rettig M (1991) Nobody Reads Documentation. *Communications of the ACM*, 34(7), 19–24. https://doi.org/10.1145/105783.105788

Rieder B and Röhle T (2017) Digital Methods: From Challenges to Bildung. In: Schäfer MT and van Es K (eds) *The Datafied Society: Studying Culture Through Data*. Amsterdam: Amsterdam University Press, pp.109–124.

Rieder B, Peeters S and Borra E (2021) From Tool to Tool-Making: Reflections on Authorship in Social Media Research Software.

Ruppert E and Scheel S (2019) The Politics of Method: Taming the New, Making Data Official. *International Political Sociology*, 13(3), pp. 233-252.

Sapkowski A (2008) *The Last Wish*. Little, Brown & Company.

Tenen D (2016) Blunt instrumentalism: On tools and methods. In *Debates in the digital humanities* 216 (pp. 83–91). https://dhdebates.gc.cuny.edu/read/untitled/section/09605ba7-ca68-473d-b5a4-c58528f42619

Van Es K, Schäfer MT and Wieringa M (2021) Tool criticism and the computational turn, a "methodological moment" in media and communication studies. *Medien Und Kommunikationswissenschaft*, 69(1), 46–64. https://doi.org/10.5771/1615-634X-2021-1-46

Van Geenen D (2020) The Role of Software Tools in Critical Data Studies Practices. Studying the Affordances of Gephi as a Sociotechnical System. In: Burkhardt M, Grashöfer K, Shnayien M, and Westermann B (eds) *Explorations in Digital Cultures: On the Politics of Datafication, Calculation and Networking*. Lüneburg: meson press.

Wieringa M (2020) What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability. *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 1–18. https://doi.org/10.1145/3351095.3372833