Particles of a Whole: Design Patterns for Transparent and Auditable AI-Systems

Judith Fassbender
Alexander von Humboldt Institute for Internet and Society
Berlin, Germany
judith.fassbender@hiig.de

ABSTRACT

The integration of principles such as transparency and auditability in AI-systems is a challenging endeavour. The demand for guiding concepts is to be not too high-level and not too specific at the same time. In this paper, I will introduce design patterns as a promising method to meet this demand and as a guidance for the implementation of transparency and auditability in sociotechnical AI-systems. This will include the history of design patterns in urban planning and software engineering, as well as their distinguishing features. In extension to this, I outline a pattern approach for responsible AI-systems in a democratic society. This includes a sketch of possible design patterns for transparent and auditable AI-systems. Some early challenges and open questions are discussed at the end, together with the conclusion.

CCS CONCEPTS

• Computing methodologies → Artificial intelligence • Software and its engineering → Software creation and management • Human-centered computing → Collaborative and social computing • Social and professional topics → Computing / technology policy

KEYWORDS

Design Patterns; AI; Auditability; Transparency; Democratic Processes

ACM Reference format:

Judith Fassbender. 2021. Particles of a Whole: Design Patterns for Transparent and Auditable AI-Systems. **In *Proceedings of ACM UbiComp/ISWC' 21.***

UbiComp-ISWC '21 Adjunct, September 21–26, 2021, Virtual, USA

ACM ISBN 978-1-4503-8461-2/21/09.
https://doi.org/10.1145/3460418.3479345

1 Introduction

While there is growing agreement on ~~the~~ responsible AI principles such as transparency and auditability, moving from these principles to actual implementation is quite challenging. Principles are in general too high-level; and case studies are generally too specific; to be reused by other researchers and developers of AI-systems. Design patterns, an idea introduced in urban planning by Christopher Alexander in 1977 [2], and adopted in a variety of other domains, including software engineering (since the 1990s) [4, 6, 7], may be able to fill this gap. In this paper I shall elaborate on this idea, and present examples of such possible patterns. A design pattern is a description of a solution to recurrent problems [4, p. 10]. These descriptions have a distinct structure; a problem, context, solution, for instance [1, p. 247].

Singular design patterns provide a vocabulary and a mental structure for the implementation of the solution and exchange of ideas. Repositories of patterns further allow the integration of a moral capacity which is shared by all patterns. In urban planning, this is the capacity to produce a living structure that is generative, and leads to people who use the space having agency. In software engineering, this shared core has been less explicitly expressed, but can perhaps be best explained as producing software systems that are easily adaptable [4, p. 11ff.].

If we are to identify design patterns for transparent and auditable AI-systems, this common capacity would involve unpacking the 'sociotechnical' nature of such systems. We desire transparency and auditability in AI-systems to protect democratic ideals and avoid authoritarian or inegalitarian societal outcomes. In other words, the moral capacity is the democratic norm, or stated differently: transparency and auditability are desirable to be accountable to (among others) the general public.

In the rest of this paper, I will explore two example design patterns for enhancing transparency and auditability in AI-systems. The paper is organised as follows. In Section 2, I will present the history and specifics of the design pattern approach; in Section 3, I outline a possible pattern language for responsible AI-systems. In Section 4, I sketch two preliminary patterns, which I have been developing together with a computer linguist and a

computer vision expert. In Section 5, I discuss some early challenges and open questions and conclude.

2 Background

In this section, I will briefly go through the importance and history of pattern approaches, describe why they are promising for responsible AI-systems and what features of the structure have to be altered to focus on transparency and auditability.

2.1 History and Importance of Pattern Approaches

Design patterns are a format that captures solutions to recurring problems. Christopher Alexander, to whom design patterns go back to, describes design patterns as follows: "Each pattern describes a problem which occurs over and over again […], and then describes the core of the solution to that problem, in a way that you can use this solution a million times over, without ever doing it the same way twice." [2, p. 10].

Whilst patterns go back to urban planning, they have been used extensively in computer science. The approach inspired the structure of Wikipedia [3] and a variety of pattern catalogues [4, 6, 7] have been published in the field of software engineering.

The pattern approach has been used merely as a knowledge management tool in software engineering, to capture collective knowledge, structure it and to share it.

Gamma et al. [4] presented a seminal set of 23 design patterns, for object-oriented programming. The patterns are structured in the sections "Intent", "Also known As", "Motivation", "Applicability", "Structure", "Participants", "Collaborations", "Consequences", "Implementations", "Sample Code", "Known Uses", "Related Patterns" [4, p. 17f.]. Most sections are described in several paragraphs, which results in a fine grained overview on how to apply the described principles. Lakshmanan et al. [7] provide a catalogue for design patterns for machine learning, with a focus on computational aspects. Whilst they integrate a few patterns which are summarised as 'Fairness Lens', they do not explicitly understand AI-systems as sociotechnical systems. Both examples from software engineering are well-received sources for the transfer of knowledge, but have a different focus than the original approach in architecture, which emphasised the importance of social events.

Alexander et al. [2] look at technical architectural aspects without neglecting social events and structures which are connected to the physical surrounding. The developed patterns concern construction aspects (e.g. 194. interior windows; 211. thickening the outer walls), as well as patterns of social encounters (e.g. 18. network of learning; 40. old people every-where; 57. children in the city) or mobility (e.g. 11. local transport areas; 22. nine percent parking)[1]. The patterns are part of a whole pattern language. This means they cannot be understood as fragments but have to be combined, to find a solution for the design problem at hand, which supports the creation of a living structure and agency in every aspect. Their approach is particular and holistic at the same time.

Software design patterns focus on the architecture of a system but not on the social processes, which take place within the (software) buildings they help to create. As I have argued in the beginning, it is necessary to understand AI-systems as sociotechnical in nature and ground them on a democratic norm and therefore integrate democratic processes, to ensure a rigorous implementation of transparency and auditability.

The design of digital infrastructures increasingly affects human interaction, and can therefore support or hinder certain social events. Alexander's statement about spaces and events can be translated to AI-systems in social contexts: "[A] pattern of events cannot be separated from the space where it occurs." [1, p. 73]. The pattern of an automated decision (which can be the cause for an event) cannot be separated from the design of the systems which caused that decision.

2.2 Why Patterns are Promising for Auditable and Transparent AI

To summarise, pattern approaches have several benefits for a rigorous integration of transparency and auditability in AI-systems:

1. Patterns serve as **a tool for knowledge management**, as they present complex content in a neat and coherent structure. When used in an open fashion, pattern approaches support a collective use and provide vocabulary [4, p. 24]. Besides enabling an exchange, this vocabulary can be understood as cognitive support and can "help you identify less-obvious abstractions and the objects that can capture them." [4, p. 24]. The mid-level of abstraction which is characteristic for patterns further allows the integration of the described processes in different use contexts, while at the same time offering hands-on guidance, comparable to a recipe.
2. As exemplified, pattern languages have an inherent **capacity to integrate moral norms** for a value-sensitive design, e.g. to support transparency and auditability. This foundation holds the pattern repository together and is a conceptual north-star.
3. Building upon the two previous points, design pattern approaches are a coherent format that can function as a **frame which allow research and best practices that already exist to be integrated** while aligning with the moral capacity of the pattern repository. Thus it is for example imaginable to formulate a pattern on the basis of Gebru et al. [5] 'Datasheets for Datasets'. The modular character further allows not only collective use, but also a collaborative continuous development. For this aspect, it is important to think of patterns not as fragments, but as a holistic system.

[1] https://www.patternlanguage.com/archive/ieee.html

The contested suitability for knowledge management, in combination with the possibility to ground the idea of a sociotechnical AI-system in democratic norms, distinguishes pattern approaches from other approaches to guide the implementation of transparency and auditability aspects in AI-systems. As it is becoming more clear that responsibility has to be given to a general public[2], measures to do so have to be integrated in the development and auditing process. Following this norm, patterns in their governing role can guide the actual implementation of according processes.

3 A Pattern Approach for Auditable and Transparent AI-Systems

To develop a pattern approach for auditable and transparent AI-systems, different components of pattern approaches have to be altered: the singular patterns have to fit the needs of practitioners as well as the item the pattern deals with – as described for the patterns by Gamma et al. [4] earlier. The repository of interlinked patterns has to be organised in a clear fashion and share a common norm, which defines what quality the implementation of the patterns should result in. In prominent examples from computer science, it is easy adaptability. It has to be possible to combine patterns and form individual sequences of patterns that are tailored to solve the unique design problem according to the rationale of the respective pattern language.

As touched upon in the introduction, for transparent and auditable AI-systems, the moral capacity will be democratic norms and integrating the general public. Processes that support transparency and auditability shall be captured in patterns for each step along the AI-lifecycle. In practice, the patterns can then be combined to support the realisation of the development and maintenance of AI-systems in a transparent and auditable manner. Such individual combinations offer specific guidance, while being context-sensitive and aim at a holistic solution. The shared structure of the patterns will be developed along a prototyping process and is not lined out yet.

4 Sketches of Pattern Scope

To flesh out the theoretical concept, I will share two sketches of patterns which I am developing in the research group Public Interest AI at the Alexander von Humboldt Institute for Internet and Society as part of my PhD project. One example pattern is captured in collaboration with a computer vision expert, the other example is developed with a computer linguist. Both examples have to be seen as initial sketches. The PhD project will concentrate on such singular patterns, as well as a structure to connect patterns along the entire AI-lifecycle. My main goal of presenting these examples is to show the scope of patterns I have in mind.

4.1 A Pattern for Transparent Collective Data Annotation

The pattern concerns the data pipeline, particularly the selection of labels for data annotation. The process is currently developed for an image data set of potentially barrier-(free) environments, for people who are dependent on wheelchairs. Automated image recognition of aids and barriers shall be implemented in the mapping service wheelmap.org. In this case, the labels will be defined in a workshop with representatives of their user-base.

The pattern deals with the question “Who is deciding which specific features are labelled and respectively recognised?”. If this step is undertaken by technical experts and on the basis of conceptual assumptions the risk of misconceptions of user groups increases. The pattern suggests a participatory labelling process: Representatives of the users and/or affected parties should be included in the initial identification of labels.

This process supports the transparency and responsibility of the system by opening it to participation, the representatives do not only shape parts of the system but also gain relevant insight, which can further be communicated publicly.

A further benefit of this pattern is the opportunity for the developing team to gain domain insights from users, which will improve the whole product in terms of usability and responsibility. It is further likely that the sensitivity of the technical team for the needs of their user group will improve through direct interaction and participation.

4.2 A Pattern for Validating the Accuracy of an Automated Task

An important question concerning many publicly accessible AI-systems is: “who has the expertise and ability to validate the accuracy of automated tasks”. This question is investigated in the context of a computational linguistic project in our research group, which contributes towards the automated translation of standard German text to simplified German. In this specific case, the people who will be confronted with a translation can validate the accuracy of the translation within the scope of their ability: as the application is an aid and shall serve users who depend on a simple version of a text, further validation could be needed. A translation that appears to be grammatically and orthographically correct might, for example, still be semantically incorrect. A validation through the target group is, in this case, not only defined by their technical expertise, but also by the conception of the target group. We, therefore, have multiple levels of validation of the task involving different parties (such as advocacy organisations, public bodies, and citizens) in mind.

---

2 The public is be understood conceptually and encompasses more than just civil society. It is not only an important question in practice who the public is but also a widely discussed question in political philosophy. Who is considered the general public is context dependent and will be dealt with on the level of singular patterns.

5 Discussion & Conclusion

It has been shown that design patterns have a long standing history in software engineering, as well as in sectors which are engaged with community building and maintenance such as urban planning. I further laid out that pattern approaches have proven to be an important tool for knowledge transfer, that the approach has the capacity to incorporate a moral or normative capacity, as well as integrate existing research and collective knowledge. Based on these properties I have suggested that a pattern approach is well suited to guide the realisation of auditable and transparent AI-systems. This is based in particular on the possibility to ground auditability and transparency via the moral capacity of pattern approaches in democratic norms and therefore direct responsibility towards a general public.

The outline of such a pattern approach, which is based on the integration of democratic processes, has been laid out, in which the integration of a general public in the auditing and development process are understood as key processes.

Two exemplary patterns which have been developed in collaboration with a computer linguist and a computer vision expert, have been presented. Finally, the occurring challenges and open questions at this initial stage of the project have been discussed briefly.

In this very initial phase some challenges beside the general questions start to crystallise.

An urgent question, to ensure that the patterns actually serve practitioners is: What shared pattern structure supports the usability of patterns in practice? Concerning the content of single patterns, it is a challenge to hit the sweet spot between a wide applicability and the persistence of actual guiding capacity. A pattern which would derive from the example on data annotation e.g. should be described in a way that it is applicable to different kinds of datasets and different use-contexts. Differences might be the availability of representatives, or massively differing opinions between representatives. Other formats of data than images might be less easy to understand for lay people etc..

Pattern repositories thrive from their balance in variety and order. The two outlined patterns are only two drafts of possible patterns for responsible AI. Gamma et al. [4] presented 23, Lakshmanan et al. [7] 30 and Alexander et al. [2] 253 patterns in their catalogues. A pattern language for responsible AI would ideally present a pattern repository which covers each step of the AI-lifecycle concerning deliberative design and open validation. The patterns further have to be applicable to different use-cases, which might be balanced by quantity of patterns or the degree of abstraction in singular patterns. The necessary collaboration between different groups of experts and the orientation of patterns towards collective knowledge, raises the question “How can we engage an interested expert community in the recording of patterns?”.

Some additional questions to deal with in the future include how to enhance the accountability of AI-systems with patterns; how any identified patterns can be evaluated; and what structure design patterns need to be understandable and useful to the community in practice.

ACKNOWLEDGMENTS

This research project is funded by the German Federal Ministry of Education and Research (BMBF).

REFERENCES

[1] Christopher Alexander. 1979. The Timeless Way of Building. Oxford University Press.

[2] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. 1977. *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press.

[3] Ward Cunningham and Michael W. Mehaffy. 2013. Wiki as pattern language. In *Proceedings of the 20th Conference on Pattern Languages of Programs (PLoP '13)*. The Hillside Group, USA, Article 32, 1–14.

[4] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, and Booch Grady. 1994. Design Patterns: Elements of Reusable Object-Oriented Software. 1. Edition. Addison-Wesley Professional.

[5] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, und Kate Crawford. 2020. „Datasheets for Datasets“. ArXiv:1803.09010 [Cs], März. DOI:http://arxiv.org/abs/1803.09010.

[6] Erik G. Nilsson 2009. „Design Patterns for User Interface for Mobile Applications“. Advances in Engineering Software 40 (12): 1318–28. DOI:https://doi.org/10.1016/j.advengsoft.2009.01.017.

[7] Valliappa Lakshmanan, Sara Robinson, and Michael Munn. 2020. Machine Learning Design Patterns. O’Reilly Media, Inc.