Toward a Method of Visual Artifact Analysis: Understanding Learners' Design Activity in a Makerspace

Anders I. Mørch [1] and Renate Andersen [2]

[1] University of Oslo, P.O. Box 1092 Blindern, 0317 Oslo, Norway
[2] Oslo Metropolitan University, P.O. Box 4 St. Olavs plass, 0130 Oslo, Norway

Abstract

In our research, we study end-user development and computer-supported collaborative learning in educational settings. The main research method we use is interaction analysis (IA)—the analysis of group interaction (verbal dialog) as it unfolds in real time among students—scaffolded by teachers and mediated by artifacts. IA does not focus on the dynamics (development and modification) of technological artifacts; instead, the emphasis is on gesture, deixis, and action descriptions. We argue that IA is insufficient for understanding design-intensive collaborative learning, especially in settings involving makerspaces and programming in school. Visual artifacts play an important role in computer science, engineering, and architecture, serving as representations (e.g., computers to be programmed, machines to be repaired, houses to be built, etc.). We argue that design-intensive collaborative learning needs methods for understanding both verbal and visual artifacts and we propose the first version of visual artifact analysis. The paper's main aim is to provide an argument for the usefulness of this method by providing a small example and surveying relevant literature.

Keywords

Design, end-user development, interaction analysis, visual artifact, visual artifact analysis

1. Introduction: Rationale for visual design method in educational research

In the 1980s, a team of researchers in MIT's architectural design theory and methodologies group proposed the concept of design game for architectural education [9, 10]. One of the games, called silent game, was described as follows: "The game is played by two players. One party arranges a few objects on a board (in the example these objects are nails and washers) and the other party must continue the arrangement by adding and must try to be true to the patterns implied by what the first player did. The players are not allowed to talk." [10].

Mark Gross, one of the members of Habraken's research team explained the concept in a conversation [8] focusing on the role of rules to represent patterns in technology design and group organization. Gross said the "basic message of Habraken is that rules (about the selection and position of building components) can coordinate group design. That is, designers, by making and following systematic agreements (rules) about the selection and position of elements, can work more effectively as a team. The idea that rules and systematic procedures can help a designer work effectively often meets with a great deal of distrust and resistance in the architecture profession. However, in real building practice the methods have been successful. Habraken looks at the systems of components that must be integrated in a building. There is a hierarchy of dependence among them: e.g., windows are mounted in walls. If well-formed rules have been articulated for the deployment of components, then design alternatives can be evaluated." [8]. A reason for the distrust among practitioners could be that the rules were reminiscent of the rules comprising knowledge-based (AI) systems, thought to be rigid.

Proceedings of CoPDA2022 - Sixth International Workshop on Cultures of Participation in the Digital Age: AI for Humans or Humans for AI? June 7, 2022, Frascati (RM), Italy
EMAIL: anders.morch@iped.uio.no (A. I. Mørch); renatea@oslomet.no (R. Andersen)
ORCID: 0000-0002-1470-5234 (A. I. Mørch); 0000-0002-1206-2140 (R. Andersen)

We have taken inspiration and applied these ideas in a method for analyzing visual artifacts in educational settings, particularly the idea of evaluating an artifact by decomposition to identify design elements and recreating the steps of its composition by a set of rules organized hierarchically, in effect defining an artifact's design space by a graphic language. By a graphic language we mean a form of communication achieved by composing visual images instead of sequencing words and utterances.

Figure 1 below is a verbal protocol of three pupils who work together to solve a mathematics assignment (introduction to probability) in a makerspace, using a micro:bit controller and a block-based programming language (MakeCode) to program the controller [13]. The task is to simulate a die by combining hardware and software components and where gameplay (chance) represents aspects of probability. The solution created by the group is shown at the end of the episode (adapted from [3]).

Turn	*Person*	*Utterance ((action on objects and body language))*
1.1	Sophia	It's just like… Can the Microbit be used as a die?
1.2	Olivia	Nothing more than that?
1.3	Sophia	No, that's all! ...
1.4	Olivia	Should we try making a die then, or what?
1.5	Liam	Okay. Mm. We could just do what we did earlier.
1.6	Olivia	Like… On [Microbit] shake… ((picks purple block from pallet)) Then...
1.7	Liam	Then you can add "show number" … ((Olivia picks blue block and puts inside the purple block)) and a... And maths of course. ((Olivia goes to the maths pallet and searches)) Then we have a random number ((Olivia finds the "pick random" block set to default values 0 and 10)) from 1 to 6 or 1 to 10 ((Liam sees the number 10)), but 1 to 6 ((Olivia changes the value from 10 to 6 in the innermost block)). And this should work like a die.
1.8	Olivia	((Tests Microbit by clicking the "shake" button 5 times)) It works! ((tone of voice is happy))

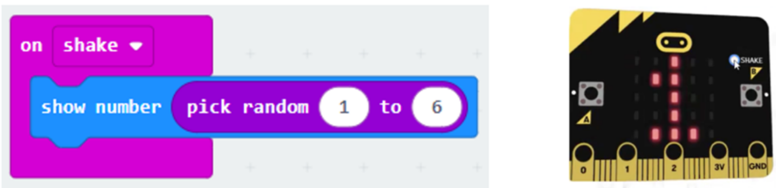

Figure 1: Interaction analysis transcript of a conversation between three pupils occurring while designing a visual artifact (software and hardware). The resulting visual artifact is composed of four design units that simulates a die when the user clicks the shake button on the micro:bit controller. Text in brackets is explanatory or inaudible and double parenthesis describes action on objects [3].

Now imagine playing the video of the episode without the sound. This allows the researcher to focus on the development of the visual artifact in incremental steps and visualizing a larger design space. We identified the different relations between the design units (DUs) that the pupils used (intentionally or implicitly) to form a four-DU assembly. This information is not fully articulated in the pupils' talk because talk is often incomplete, or the actors do not fully verbalize their design decisions. Mapping a broader design space with a brief rationale based on the rules can be useful.

In Figure 2, the design process is organized into steps of assembling the visual artifact by a series of snapshots from the data analysis video and reproduced here with the MakeCode blocks editor [13].The second column shows intermediate stages (subassembly), the third column the changes that were made at each step, and the fourth column a rule description. At each step, the pupils have some degree of freedom to choose what blocks to combine and what values to change, but the design process is constrained as we explain below. Inspired by design games, we define a composite design element (subassembly) as the result of composting DUs by selecting rules and sequencing DUs hierarchically. The researcher identifies the rules by analyzing the artifact, but the pupils may not be aware of them. The interaction analysis (IA) transcript (Figure 1) provides clues to the pupils' knowledge. The visual artifact table extends the pupils' verbal understanding by capturing a broader design space, that is, including those options that were not chosen or ignored by the pupils. In this way, visual artifact analysis provides a degree of objectivity to the otherwise subjective conversations engaged in by a group of pupils when they design together in an educational makerspace. To avoid very long visual artifact tables when analyzing complex artifacts, we suggest focusing on interesting areas, which we define as those areas in a design space where domain-specific and generic code blocks interact. The rationale for this is explained in Section 2.1.

Step	*Visual artifact (subassembly)*	*New/modified component*	*Rationale/rule*
1	on shake	show number 0	Trigger execution after shake object, showing a number on the microbit
2	on shake / show number 0	pick random 0 to 10	The number is set to an arbitrary value within a default range 1-10
3	on shake / show number pick random 0 to 10	pick random 1 to 6	The numbers 0 and 10 are modified to 1 and 6 to model the numbers of a die
4	on shake / show number pick random 1 to 6		

Figure 2: Visual artifact table. At each step a rule is invoked by connecting two design units, implying a relation between them. Step 1 is different from Step 2 and Step 3 in that the former invokes a generic rule while the latter invoke domain rules. The domain rules refer to probability and game play, respectively.

2. Related work

2.1. Visual artifacts in education

A pioneer of pedagogy and educational research in the Nordic countries is Helga Eng. She has written significant volumes on the role of drawings in human mental development from childhood to adolescence. Eng used the method of collecting drawings from the same individual over an extended period, from age 9 until 24 [6]. Eng was interested in mental development on multiple levels: first, how young adults' color drawings incorporate aspects of more general information inherited from European visual arts (i.e., how visual arts that have evolved over a long time in a society can reappear in individuals' drawings over a shorter timespan) and, second, how drawings connect with abstract concepts learned in school and applied in leisure and sports activities. Eng's analyses suggest that drawings develop from schematization (crude rendering) to organic forms (more sophisticated rendering) and major changes take place at certain ages or with the appearance of significant events in life [6].

Eng inspired our work in terms of the developmental driving force in combining domain-specific and general information in visual artifacts. In our case, this occurs when a pupil starts designing by choosing general design elements such as the two blocks shown in Step 1 of Figure 2 ("On shake" and "Show number") and next adds a domain-specific element ("Pick random" in Step 2 or setting the range "1–6" in Step 3). The domain-specific elements create a tension or contrast in the two-block design (contrasting color, shape, layering, symbol, functionality, orientation, etc.). We conjecture that juxtapositioning contrasting design elements can fuel development by encouraging domain knowledge to appear in the pupils' conversations, thus supporting collaborative learning. As opposed to this, when a generic element is added to a domain-specific element or subassembly or when a domain-specific element or subassembly over time has been internalized (taken for granted or used as a multipurpose building block), we refer to the composite element as a stable intermediate form, serving as the foundation for further development [14]. The quality of a stable intermediate form (SIF) ranges from fragile to durable, and techniques for harnessing fragile SIFs to sustain development into complex visual artifacts is an area for further work; for an example see [12].

2.2. Visual artifacts in design: Automated vs. assistive rules

The rules for design composition we have demonstrated are inspired also by shape grammars—an innovative design approach introduced in architectural design in the 1970s [19]. Shape grammars are both descriptive and generative. The rules of a shape grammar generate designs, and the rules are descriptions of the forms of the generated designs. Rules define state changes for how to get from one stage to the next, and rule invocation takes place in a work area where the stages of visual artifacts are displayed. Our rules do not form part of a grammar or a system for generating visual artifacts automatically. They are standalone and displayed in a table format and used by researchers or students to evaluate hardware (e.g., a sensor-based alarm) or software (e.g., a simulation or user interface) artifacts. Educational researchers can use the method to study collaborative design in a design-game like setting [9, 10]. A long-term goal is to teach pupils design thinking skills and end-user developers to build rule-based virtual assistants to take on the role of a substitute teacher, which we profile in another paper at the workshop [1]. Researchers in architectural design created shape grammars to automate floor plan layout of houses and structural refinement in urban development [19]. Interestingly, the shape grammar approach is perhaps not best known for creating new architectural designs but to appreciate the existing designs, such as the Palladian grammar for reconstructing a famous Venetian villa (Villa Malcontenta), consisting of 69 rules of classical art, which are applied throughout the eight stages [20].

An application of shape grammar to human-computer interaction and visual interface design is its formal use in medical diagnosis by using a visual system by Bottoni and colleagues [4].Their approach is referred to as visual rewriting system, which supports the evolution of a computer simulation by defining pictorial state changes using visual rewrite (before–after) rules. The rule-based approach to developing visual artifacts has been applied to end-user development (EUD) by Costabile and colleagues [5]. In EUD, users are the agents of change who make modifications to IT systems and create artifacts themselves, rather than the system doing it automatically. With EUD, the focus has shifted from artificial intelligence (AI) to human-centered AI (HCAI), enabled by domain-expert users who set their own goals and use advanced tools to assist task completion, relying on the computer to automate general goals and tedious tasks [1, 5, 7].

2.3. AI vs. human-centered AI

The title of this workshop is "AI for Humans or Humans for AI?" Visual artifact analysis is indirectly related to AI through knowledge-based rules in architectural design. The rationale for this line of development in our work is the aim and concern of architecture to contribute to human-centered artifacts, which we use as an ideal for HCAI and AI for humans.

The difference between the shape-grammar approach [19, 20] and design game approach [9, 10] toward visual design and rule invocation can be compared with the difference between AI and HCAI. Artificial intelligence (ranging from rule-based systems to neural networks) is about creating automated systems whereas HCAI turns AI around to intelligence augmentation. HCAI researchers develop systems to augment human tasks to help people do things better and together. We follow Fischer (this workshop) who argues for combining AI and HCAI [7]. We suggest combining two types of rules for visual artifact analysis—automated (implicit, tacit) and non-automated (explicit, assistive).

Visual artifact analysis has been prompted by the rise of programming in K–12 education. Previous research has suggested that programming can be used as an exploratory design method for learning science topics [18]. Visual artifact analysis as a research method addresses a shortcoming with IA for understanding design-intensive collaborative learning, which occur in educational makerspaces and programming tasks.

3. The interaction analysis method

We have used IA [11] to analyze empirical data in previous projects (Figure 1 shows a simple example). Interaction analysis is a social science method for empirical investigation of the interaction between humans and objects in their environment and is a useful method for analyzing the micro utterances, gestures, tone of voice, the chronological and spatial temporality of the different participants' utterances, how they are connected and take different turns, and the use of objects during interactions [11]. These objects are referred to by deictic references (this/that; here/there; now/later/before, etc.). Finally, the actions performed on the objects are captured by researchers' comments (comments are inside double parenthesis in the conversation shown in Figure 1).

The physical aspects of communication (gesture, deixis, action descriptions, etc.) are important information when analyzing learning activity in a makerspace, but what is missing is a method for capturing step-by-step visual artifact development with the lenses of a graphical or design language. Visual artifacts in learning activities are not static referents but a moving target or continuously evolving, providing a dynamic context for understanding pupils' talk [3]. In previous work we have used IA in design-intensive collaborative learning in the following ways: 1) separating analysis according to levels of abstraction (general and specific) [2, 12, 15]; 2) investigating techniques such as working around, appropriation, and using technology in new ways [16]; and 3) empirical investigation of the interdependency of discursive and technological objects [3].

4. Toward a method of visual artifact analysis based on rules

4.1. Rule types and invocation modes

In educational makerspaces, designs will typically involve hardware (e.g., microcontrollers, sensors, actuators, etc.), software (visual code blocks), and abstract elements (relations between DUs) (see Figures 1 and 2). A design can, for example, consist of three code blocks, a micro:bit controller, and four relations, as shown in Figure 1. In the further work we refer to relations by rules because relations are invoked by rules. Following Schön who proposed rules for designers to analyze designs that derived from types and modeled after design-like practice, such as employing tacit and explicit knowledge and general and specific reasoning [17], we have organized our rules into automated/implicit and assistive/explicit and three invocation modes: automated, generic, and domain-specific:

- *Automated rules* are implicit (taken for granted, tacit, automated, e.g., some blocks snap together) and general, which means they are applicable across multiple domains and tasks. Sequence and hierarchy are the overarching principles for this rule type, relying less on human interaction.
- *Generic rules* are distinguished from automated rules by requiring human attention (learners must be consciously aware how two design units are combined using these rules) and have in common with automated rules the property of being general (domain independent).
- *Domain rules* are distinguished from the first rule mode by being explicit (requiring human attention) and the second by being domain-specific (pertaining to a specific domain or task).

The combination of automated and assistive rules leads to a constrained design space, that is certain things are allowed and other things are prevented, which has both strengths and shortcomings. The strength is that the design elements snap together to create syntactically correct programs, which is good for novice programmers by making learning easier. The shortcoming is that programmers have fewer alternatives to choose from during composition because they are not allowed to make syntactical errors and learn from it. The shortcoming can be counterbalanced by a large repertoire of design elements to choose from (palette of parts) and allowing users to save subassemblies in the palette for later use and extend the repertoire. This identifies areas for further work.

4.2. Outline of a method

We suggest the following steps for analyzing a visual artifact's development trajectory:

1. Defining the rules and identifying the design space for a set of components.

2. Reverse engineering (decomposing) the artifact into the steps of its composition (as shown in Figure 2).
3. Invoking for each step a rule that defines a relation between two adjacent design units to justify the transition toward a more complex subassembly.
4. Comparing the user-created design/subassembly at each step with the alternatives that could have been created and using this information together with the verbal transcript to discuss the pupils' design decisions (they were good, incomplete, missing, dominated by some, etc.).
5. Determining the options users can choose from to continue developing the artifact; or, in other words, why did they stop at this stage?

5. Discussion and open issues for further work

In this position paper, we have argued for a new research method for analyzing learners' design activity in educational makerspaces. The method is based on interaction analysis, extending it with a "graphic language" for understanding design-intensive collaborative learning, that is, learning activities that involve building and modifying visual artifacts in educational makerspaces and programming assignments in a step-by-step manner. Related work inspired our approach, in particular rules in architectural design (we suggested different types and modes from design research and indirectly from symbolic, rule-based AI), transitions in design space analysis (we suggested a table format for display, adopted from qualitative research methods), and juxtapositioning generic and domain-specific design elements (to narrow complex artifact tables toward interesting areas, adopted from research in education). The current work is preliminary and should be regarded as a working hypothesis, which needs to be tested and refined to be useful in practice. We plan to use the method in an ongoing project (ProSkap) by identifying those parts of a visual artifact that can provide insights into pupils' collaborative knowledge creation and we currently seek to find this information in the intersection of technological and discursive object trajectories.

Another direction for further work is to investigate methods in the design sciences that are comparable with visual artifact analysis as advocated in this paper. Visual artifacts play an important role in computer science, engineering, and architecture, serving as representations (e.g., computers to be programmed, machines to be repaired, houses to be built, etc.). Researchers in these fields have developed many different methods for design and analysis. We will investigate some of these methods in further work, according to their usefulness and usability, in particular the extent to which these methods can be used by people who are not trained in formal design science (engineering and CS), that is, researchers in education and learning sciences who are our domain-expert users. Model-based approaches seem relevant, where a model (e.g., state machines) provides an explicit description of the user interface behavior of an application system. However, these models are based on formalisms from theoretical computer science (e.g., finite state machines) and may require adaptation for domain-expert users. A key tool is the visual model editor, and application areas include game design [21].

6. Acknowledgments

The ProSkap (Programming and Making in school) project is funded by Regional Research Funds (RFF) of the Research Council of Norway. We thank Kristina Litherland for contributing to the research presented here.

7. References

[1] R. Andersen, E. Gjølstad, A. I. Mørch, Integrating human-centered artificial intelligence in programming practices to reduce teachers' workload, in: Proceedings of the 6th workshop on Cultures of Participation in the Digital Age (CoPDA 2022), international workshop at AVI 2022, Frascati, Rome, Italy (2022).

[2] R. Andersen, A. I. Mørch, Mutual development in mass collaboration: Identifying interaction patterns in customer-initiated software product development, Computers in Human Behavior 65 (2016) 77–91.
[3] R. Andersen, A. I. Mørch, K. T. Litherland, Learning domain knowledge using block-based programming: Design-based collaborative learning, in: Proceedings of the 8th International Symposium on End-User Development, IS-EUD 2021, volume 12724 of Lecture Notes in Computer Science, Springer, Cham, Switzerland, 2021, pp. 119-136.
[4] P. Bottoni, G. Mauri, P. Mussio, G. Pun, Computing with shapes, J. Vis. Lang. Comput. 12 (2001) 601–626.
[5] M. F. Costabile, D. Fogli, P. Mussio, A. Piccinno, Visual interactive systems for end-user development: A model-based design methodology. Trans. Sys. Man Cyber. Part A 37 (2007) 1029–1046.
[6] H. Eng, Barne og Ungdoms-Tegningens Psykologi: Fra år 9 til 24. Cappelen, Oslo, Norway, 1944, English version: The Psychology of Child and Youth Drawing, Routledge and Kegan Paul, London, UK, 1957.
[7] G. Fischer, A research framework focused on humans and AI instead of humans versus AI, in: Proceedings of the 6th workshop on Cultures of Participation in the Digital Age (CoPDA 2022), international workshop at AVI 2022, Frascati, Rome, Italy (2022).
[8] Fischer, G., Nakakoji, K., Gross, M., Morch, A., Hill, W., Wroblewski, D., Terween, L., & Henninger, S. (1992). Discussion about Donald Schon's keynote address at the 1992 Edinburgh conference on Creativity and Rationale. Unpublished paper.
[9] J. Habraken, M. D. Gross, Concept design games, Design Studies 9 (1988) 150-158.
[10] J. Habraken, M.D. Gross, J. Anderson, N. Hamdi, J. Dale, S. Palleroni, E. Saslaw, M.-H. Wang, Concept Design Games: A Report Submitted to the National Science Foundation. Book One: Developing. Book Two: Playing, MIT Department of Architecture, Cambridge, MA, 1987.
[11] B. Jordan, A. Henderson, Interaction analysis: Foundations and practice. The Journal of the Learning Sciences 4 (1995) 39-103.
[12] K. T. Litherland, A. I. Mørch, Instruction vs. emergence on r/place: Understanding the growth and control of evolving artifacts in mass collaboration. Computers in Human Behavior 122 (2021) https://doi.org/10.1016/j.chb.2021.106845
[13] Microsoft MakeCode, 2022. URL: https://makecode.microbit.org/ .
[14] A. I. Mørch, Evolutionary growth and control in user tailorable systems. In: N. Patel (Ed.), Adaptive Evolutionary Information Systems, IGI Global, Hershey, PA, 2003, pp. 30–58.
[15] A. I. Mørch, R. Andersen, R. Kaliisa, K. Litherland, Mixed methods with social network analysis for networked learning: Lessons learned from three case studies, in: Proceedings Networked Learning Conference 2020, Kolding, Denmark, online, 2020.
[16] A. I. Mørch, L. Zhu, Component-Based Design and Software Readymades, in: Proceedings of the 4th International Symposium on End-User Development, IS-EUD 2013, volume 7897 of Lecture Notes in Computer Science, Springer, Berlin-Heidelberg, Germany, 2013, pp. 278-283.
[17] D. A. Schön, Designing: Rules, types and worlds, Design Studies 9 (1988) 181-190.
[18] B. Sherin, A. A. diSessa, D. Hammer, Dynaturtle revisited: Learning physics through collaborative design of a computer model, Interactive Learning Environments 3 (1993) 91-118.
[19] G. Stiny, Introduction to shape and shape grammars, Environment and Planning B: Planning and Design 7 (1980) 343-351.
[20] G. Stiny, W. J. Mitchell, The Palladian grammar. Environment and Planning B: Planning and Design 5 (1978) 5–18.
[21] M. Zhu, A. I. Wang, Model-driven game development: A literature review. ACM Computing Surveys 52 (2019) 1-32. https://doi.org/10.1145/3365000