



Behavioral Complexity as a Computational Material Strategy

Mads Hobyte* and Maja Fagerberg Ranten

Computer Science, Department of People and Technology, Roskilde University, Denmark

This paper presents the concept of behavioral complexity as a computational material strategy. The materiality of the designed interaction is a relatively new perspective on interaction design. From this perspective, the behavioral complexity should be understood as the underlying algorithms in the computational code. Complexity in the code enables multiple unique material qualities of computational materials to adapt and come to life through interaction. We propose that behavioral complexity contributes to creating expressive complexity and then present strategies of behavioral complexity as annotations in an annotated portfolio of design examples. For each annotation, simple computational programming patterns are included to illustrate practical implementations. The strategies are to create: *reactiveness, multiple modes, non-linearity, multiple layers and alive connotations*. Finally, we point towards the potential of mixing the strategies to expand the complexity of alive and adaptive expressions and discuss strategies for preserving coupling.

Keywords – Material Expressions, Behavioral Complexity, Interaction Design, Annotated Portfolio.

Relevance to Design Practice – This paper presents a set of strategies for interaction designers when working with behavioral complexity within tangible and physical computing in relationship to computational technology in order to elicit expressive complexity.

Citation: Hobyte, M., & Ranten, M. (2019). Behavioral complexity as a computational material strategy. *International Journal of Design*, 13(2), 39-53.

Introduction

Our intention in this paper is threefold. Firstly, we argue for a shared agenda around the computational complexity in a computational material. We frame this shared interest in *behavioral complexity* as a way to create *expressive complexity*. Secondly, using five design examples in an annotated portfolio, we present a set of strategies that can be used in designing alive and adaptive expressions based on behavioral complexity as inspirational building blocks. Thirdly, we discuss the potential for a strategy in mixing the strategies.

We position our work within the field of interaction design (Bødker, 2006; Fallmann, 2008; Löwgren, 2007). We focus on the computational material and how complexity in the code is part of the form-giving practice in interaction design within tangible and physical computing. We use the concept of behavioral complexity to distinguish between general code and the part of the code that intentionally affects the expression. From a designerly point of view, we are interested in how the designer/developer can explore the complexity of the computational material as a resource to create alive and adaptive designs.

Within the field of interaction design, a selection of scholars has discussed computational complexity within computational, alive and adaptive materials. These are typically presented as an element in an overarching design strategy with multiple elements at play. Gaver, Beaver, and Benford (2003) use the concept of ambiguity as a design strategy to open up the curiosity space of the participants. Through ambiguity, the interactions can be “intriguing, mysterious, and delightful” (p. 233). One way to create ambiguity is through misinformation or ambiguous responses from the system. Similarly, Tieben, Bekker, and Schouten (2011) argue for the complexity of the system as one of multiple strategies to prolong

the discovery of an installation. Hobyte (2014) argues for designing for *homo explorens* as an extension of Gaver’s (2009) designing for *homo ludens* as a way to create socially playful explorations with internal complexity. From a computational composite perspective, Vallgård and Sokoler (2010) argue for composite materials that play with our expectations of what we consider *natural*. One overlapping aspect is using computational complexity to expand the interaction space of the exploration. Larsen (2015) introduces the concept of *væsen*, a Danish word that can be loosely translated as *essence* or *animism*. His intentions are “about *actual* interactive behavior in relation to the character and role of tangible artifacts as entities with some rudimentary agency” (Larsen, 2015, p. 41). Levillain and Zibetti (2017) examine the psychological properties a behavioral object evokes in an observer. They talk about three levels of perceived complexity: *animacy*, *agency* and *mental agency*. In the simplest form, animacy is the ability to initiate and change movements spontaneously. In the most complex form, *mental agency* is the ability to display attitudes with respect to other agents (Levillain & Zibetti, 2017).

Expanding the interaction space with computational complexity seems to be challenging. Gaver, Bowers, Kerridge, Boucher, and Jarvis (2009) share their frustration about finding

Received March 28, 2018; Accepted May 1, 2019; Published August 31, 2019.

Copyright: © 2019 Hobyte & Ranten. Copyright for this article is retained by the authors, with first publication rights granted to the *International Journal of Design*. All journal content, except where otherwise noted, is licensed under a *Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License*. By virtue of their appearance in this open-access journal, articles are free to use, with proper attribution, in educational and other non-commercial settings.

*Corresponding Author: mads@hobyte.dk

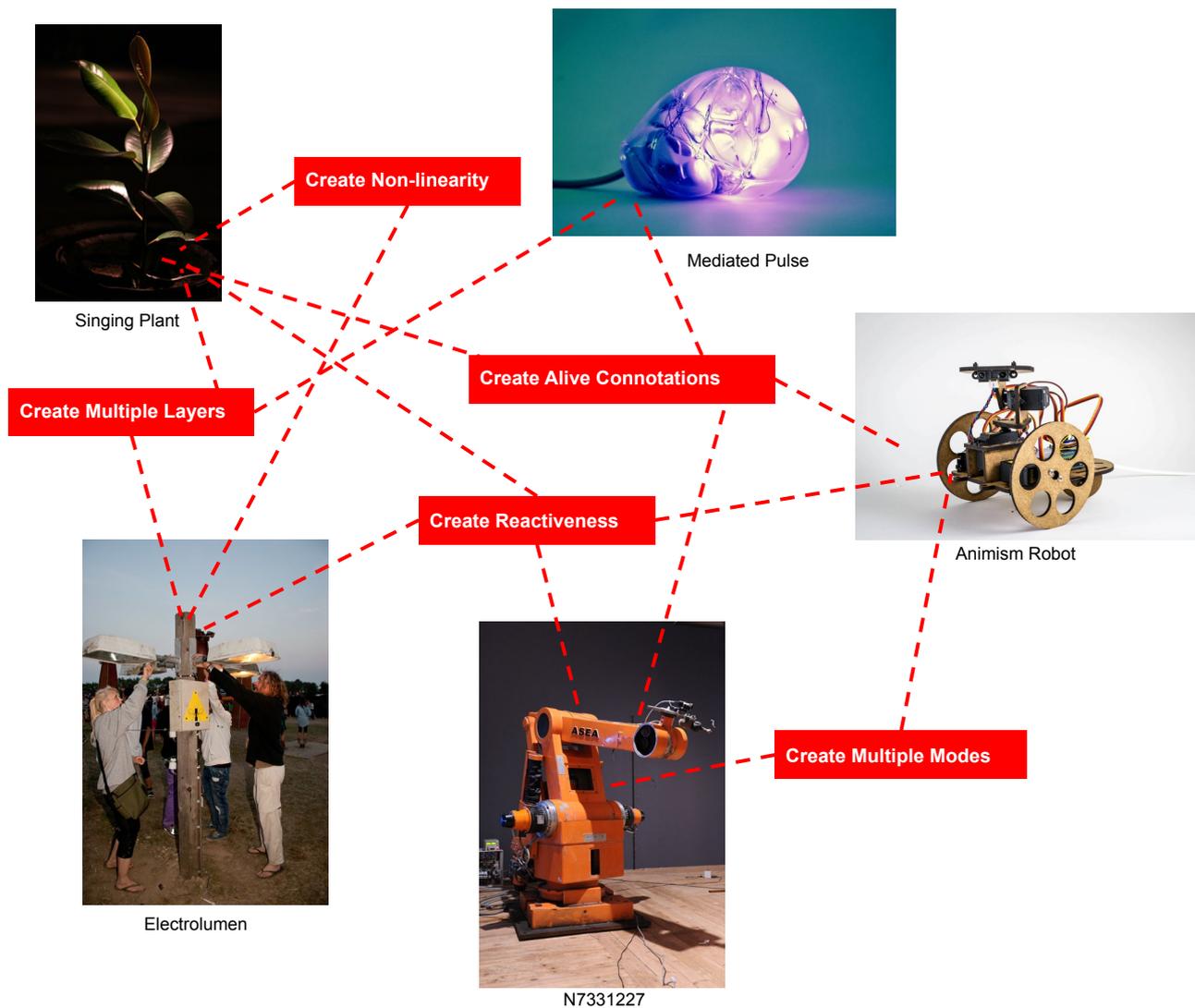


Figure 1. Annotated portfolio of examples of behavioral complexity.

the *sweet spot* between effective randomness and total accuracy in the Home Health Monitor system. Tieben et al. (2011) reflect on the limits of out-of-context disruptions and wonder if some level of complexity could elicit interactions beyond “a short spur

of curiosity and exploration before the student would be satisfied and walk on” (p. 365). With Mediated Body, Hoby (2014) presents accounts of challenging work with internal complexity for eliciting certain types of interaction.

One could see the above views as separate. However, from the perspective of computational complexity, they all point to designing code with some level of complexity in relation to evoking similar kinds of alive and adaptive expressions. We posit that there is a need to conceptually ground computational complexity for adaptive and alive expressions within the field of interaction design. We do this by presenting a model of the relationship between behavioral complexity and expressive complexity. We then suggest a set of strategies that designers can use as a starting point for exploring this practice. The strategies are presented as annotations in the portfolio and introduce the following concepts:

- **Create Reactiveness:** To create interfaces that react in real time with the interaction.

Mads Hoby holds a PhD in interaction design from Medea, Malmö University and is a co-founder of illutron collaborative interactive art studio. He conducts research into the potential of digital material exploration within art and technology. He has a keen interest in maker hacktivism and experimental electronic upcycling. As an Assistant Professor at Computer Science RUC, he has co-created Exocollective and Exostudio to create cross-pollination between artist, scientists, innovators, and makers in general.

Maja Fagerberg Ranten is an Interaction Designer and PhD Fellow at Computer Science, Roskilde University, Denmark. She is part of the Copenhagen art and technology scene and has a wide repertoire of interactive art installations from the design collaboration UNMAKE and as a member of the art collective illutron. At Roskilde University, she is co-creator of the research collective Exocollective and Exostudio, where the research focus is on digital material exploration in interactive design, art, and technology. Her PhD Designing for Bodies with Bodies is a practice-based investigation of the designers’ bodily interaction with materials when designing artistic interactive systems with a focus on an expansion of a material framework that considers both physical, computational and the body as material.

- **Create Multiple Modes:** To create multiple modes in the system that invites for different kinds of interaction.
- **Create Non-linearity:** To create internal logic without linear causality.
- **Create Multiple Layers:** To combine multiple non-linear parameters into a multidimensional interaction space for participants to explore.
- **Create Alive Connotations:** To create computational patterns with anthropomorphic, zoomorphic and/or animistic expressions.

Before we dig into the different strategies, it is necessary to ground computational complexity. In the following section, we examine in detail the interplay between behavioral complexity, physical form, computational form and expressive complexity.

Behavioral Complexity for Complex Expressions

As presented in the introduction, multiple scholars have an overlapping interest in exploring computational complexity in relation to interaction. We term this *behavioral complexity*. For us to understand the properties that are at play, we posit that it is necessary to create a conceptual model of the elements. In the following, we use the iceberg metaphor as a way to talk about the internal computational system, the physical form and the expressions as a whole.

Vallgård (2014) makes a distinction between physical form and temporal form. The physical form combined with the temporal form is what Vallgård considers the computational composite. Since the temporality of the computer and physical form “determine the temporal expression of any computational thing” (Vallgård, Winther, Mørch, & Vizer, 2015, p. 2), the internal complexity relates to the expression of the thing. Similarly, Hallnäs and Redström (2002) turn the classic Bauhaus concept of *form follows function* around and argue for the concept that *function resides in the expression* of things to revitalize the importance of the materiality of interaction design as computational things with expressions. As Hallnäs (2011) later formulates it: “As the computer disappears in the background, computational technology reappears as a new expressive design material. We build things with a new material when we build computational things, their behavior in use depending on the execution of given programs” (p. 76). This means that within computational materials, expressions have behavior and that the internal computational logic in the system plays a significant part in this.

Inspired by Vallgård’s (2014) distinction between physical and temporal form and Hallnäs and Redström’s (2002) argument for the expression of things, we introduce the visual model: Model of the relationship between behavioral complexity and expressive complexity (see Figure 2). We use the iceberg metaphor as a basis for our model. The part of the iceberg that is above the water surface represents what is visually and physically present; what is below the water surface represents what is initially hidden. What is below, only presents itself indirectly, through interacting with the system.

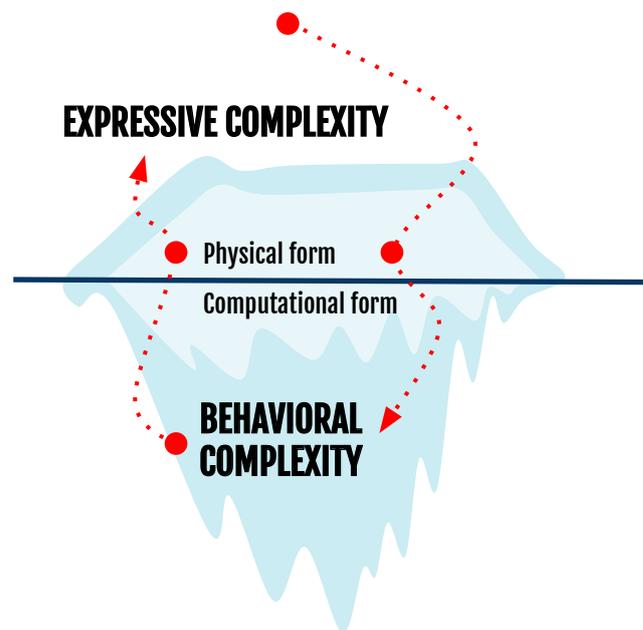


Figure 2. Model of the relationship between behavioral complexity and expressive complexity. The two arrows illustrate the dynamic feedback loop between the interactions as input that goes through the behavioral complexity to become expressive complexity.

Above is the *physical form*. This is the actual spatial dimension of the tangible object that participants can interact with, ‘participant’ referring to a user interacting with an interactive system. The tangible object consists of multiple elements: physical materials, electronics and embedded computers. The physical materials commonly wrap the electronics and a computational system into a physical form. The electronics consists of the gateways between the computational system and the physical world. Inputs are sensors: buttons, touch sensors, cameras, etc. Outputs are actuators, speakers, LEDs, etc.

Beneath the surface resides the *computational form* in the form of code. This enables the material to be alive and adaptive. This typically presents itself temporally through interaction. The combination of the physical form and the computational form provides the basis for the overarching expression of the thing or the material. From a behavioral complexity perspective, we consider the expressions as *expressive complexity*. The expressive complexity is a product of the behavioral complexity in the computational form combined with the physical form.

We are only using experiential anecdotes as a foundation for exemplifying the expressive realism of the object itself. As Hallnäs (2011) phrases it: “Expression is what makes experience possible, which is why concepts and theories of experience can never provide a logical foundation for design aesthetics” (p. 75). By using the perspective of expression instead of more experience-oriented perspectives (see e.g., Löwgren, 2002), we align ourselves with Hallnäs’ distinction of aesthetic realism as a frame to discuss the expressional logic of designed things themselves as a design perspective when creating strategies for behavioral complexity.

We argue for the concept of *behavioral complexity* to distinguish between general code in the computational form and the code that intentionally affects the expressive complexity of the computational material. Behavioral complexity is the underlying algorithms in the computational code created to enable the computational form to come alive through the physical form. We consider behavioral complexity to have the following properties:

- **Does not have simple deterministic linear temporality:** The behavioral complexity affects the expressive complexity in such a way that it cannot be considered a predefined sequence of events (like a musical score), but instead consists of internal logic that reacts to its environment.
- **Adds to the complexity of the expression:** Behavioral complexity consists of code that deliberately intends to create complex expressions. One could easily think of complex code with a rather simple expression. For example, when using artificial intelligence to detect a smile the code is complex, but the output only amounts to a binary response.
- **Has non-trivial internal complexity:** A behavior becomes non-trivial (Hobye & Löwgren, 2011) by having an internal logic that creates a set of expressions not easily apprehensible in a predictable way. As Hobye and Löwgren discuss in relationship to non-trivial internal complexity: “[I]f it is perceived as mastered easily and not complex at all, boredom will rapidly set in” (p. 46). In this sense, whether something is non-trivial ultimately resides in the experience of the participant interacting with it. However, with the concept of non-trivial internal complexity, we want to emphasize the *designerly intentions* behind the code as a part of the behavioral complexity.

The properties are intended to identify a set of prevailing strands that can enable designers to have a generative concept for considering computational form as behavioral complexity.

Annotations as Generative Definitions

Our intention is to convey a set of strategies for behavioral complexity as computational form. We do not intend to create an encompassing taxonomy, but merely to create a generative (Gaver, 2012) starting point for designers to explore behavioral complexity. We do this by presenting five technical design examples that encompass some level of behavioral complexity. This allows us to look at how the complexity of the internal code, and thus the expression of the system, can be used as a design strategy for alive and adaptive materials.

The five examples are presented as an annotated portfolio (see Figure 1). The concept of an annotated portfolio was coined by Gaver and Bowers (2012) as a designerly way to present a spatial map to exemplify a design potential and as a way to communicate design research as a form of theory formation. An annotated portfolio is, in its simplest form, a collection of design objects with textural annotations to exemplify topics or themes, indicated by the annotations where the annotations and the designs are mutually informing (Gaver & Bowers, 2012). Gaver (2012) points to a single design as a point in design space whereas

a collection—a portfolio—establishes an area in that space. The role of theory should be to annotate those examples rather than replace them.

Similarly, Redström (2017) articulates a notion of assemblages (in relation to defining researchers work with examples; a set of particular designs in combination with an overall program framing) as a meaningful whole: “It is an assemblage of definitions that aims towards a meaningful whole, not towards isolated and contained concepts. It is a hands-on way of working with, and explicitly addressing, the tension between the making of the particular and an overall orientation toward the more general through design” (p. 115).

The design examples for the annotated portfolio are a selection of pieces produced over an extended period of research. The examples vary significantly in form, size, material selection and context of use. However, they all have elements of behavioral complexity in them. Furthermore, all of them have been explored in different contexts through multiple iterations of the code. A relatively large amount of knowledge has been gained about their expressive qualities. They have been consciously selected to create a wide space for which the annotations can exemplify a more general understanding of what constitutes behavioral complexity.

We posit that the knowledge contribution of communicating design exemplars as an annotated portfolio is beneficial as it allows us to look across different experiments and projects. It allows us to take a bird’s eye view and compare, reflect on and differentiate design examples. Additionally, it allows us to generate multiple perspectives on behavioral complexity so that the following discussion can result in knowledge production with various possible actions as opposed to one particular action.

Portfolio: Design Examples for Behavioral Complexity

The following is the portfolio of the five design examples. The five examples explore different levels of behavioral complexity. We intend the examples to show the diversity of the relationship between behavioral complexity and expressive complexity. For example, Mediated Pulse’s organ-like shape gives a clear expression of organic qualities even though the code itself has a relatively simple rhythmic logic.

Electrolumen

Physical Form

Electrolumen (presented by Hobye, 2014; Hobye, Padfield, & Löwgren, 2013) consists of telephone poles, two meters tall, with four street lamps mounted just above head height (see Figure 3). Four wires, looking exactly like standard aerial high voltage electrical cables, are mounted on real ceramic insulators and go off at chest height to another telephone pole several meters away. Electrolumen gives the impression of something both well-known and very dangerous that we are used to seeing far overhead, bringing it suddenly and disconcertingly within easy reach.

Behavioral Complexity

The four lamps are touch connection sensitive, that is, touching one will not elicit any response, but touching any two simultaneously will cause the lamps to light up and the installation to generate sound. The texture of the sound is affected by how firm the touch is. The sensor electronics derive an analog input variable by whether your hand is a half centimetre away, lightly touching, or firmly gripping the lamp. Holding more than two lamps gives more light. Each different combination of four lamps gives a different set of sounds. There are six sound channels, which can be combined. Three behavioral complexity strategies have been used: Reactiveness, Non-linearity, and Multiple Layers (see Table 1).

Expressive Complexity

Electrolumen facilitates social exploration in which the participants make contact with each other. In order to light all four lamps and to play with different sound ambiances, participants need extra hands to help them. A connection can also be made through a friend, a human chain or by kissing a stranger. The design was exhibited in a festival context in which we explored how to make ambiguous interfaces (Gaver et al., 2003) elicit socially playful (Gaver, 2009) and exploratory (Hoby, 2014) interaction between the participants.



Figure 3. Participant engaging with the Electrolumen installation at Roskilde Festival.

Animism Robot

Physical Form

The Animism Robot is an explorative learning platform for creating animatronic behavior and physical expressions (Padfield, Haldrup, & Hoby, 2014). The robot consists of an Arduino microcontroller, four servos, a distance sensor and physical shapes cut out of HDF wood (see Figure 4). The motors allow the robot to drive around on a flat surface and tilt and rotate its head.

Behavioral Complexity

As a part of the kit, a few code pieces were included, which would introduce different aspects of animating the robot. One would make the head shake. Another would make the robot move forward if there were no obstacles in front of it. A vast number of extensive hacks have been done to the platform both physically and computationally. From a behavioral complexity perspective, the computational logic would be extended to create more complex patterns. For example, the robot would move forward and stop if an object were close to it. If the object started to move towards the robot, the robot would try to run away, thus creating the illusion of an optimal distance between the interactors and the robot coming from the robot itself. Three behavioral complexity strategies have been used: Reactiveness, Multiple Modes, and Alive Connotations (see Table 2).

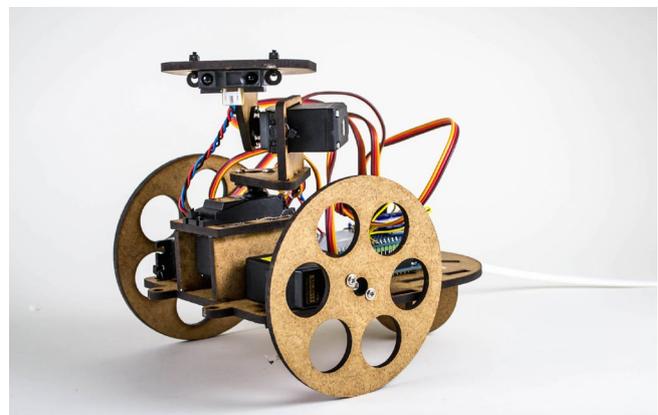


Figure 4. The Animism Robot kit presented in its assembled form.

Table 1. Three behavioral complexity strategies have been used for Electrolumen.

Reactiveness	Non-linearity	Multiple Layers
The installation uses a responsive action->reaction pattern to enable participants to play and explore the interaction dynamics in real time.	Nonlinear algorithms give a more complex and dynamic feel to the interaction, compared to the relatively simple touch interface.	Amount of touch, length of touch, pull on wires, change in touch and change in pull are all measured and converted to an <i>energy level</i> , enabling more complex patterns to emerge than a simple on/off touch interaction.

Table 2. Three behavioral complexity strategies have been used for the Animism Robot.

Reactiveness	Multiple Modes	Alive Connotations
The platform uses a responsive action->reaction pattern to enable participants to play and explore the interaction dynamics in real time.	The robot has multiple moods. It can be programmed to be happy, curious, scared, bored, etc. These different modes allow for a prolonged interaction where new dimensions of the <i>personality</i> of the robot are explored.	The code plays heavily on the social connotations of body movement and facial expressions to give the impression of the robot being <i>alive</i> .

Expressive Complexity

The robot provides the participants with the possibility to think, experiment and express themselves through the material, the media of the behaviors, the look, feel and interaction of a robot (Hobye, 2014). Through the many hacks of the code and the physical configuration of the robot, many different behavioral qualities have been designed. They have explored different abbreviations of animism (Larsen, 2015) and how we as humans start to relate to mechanical objects as having an embedded personality or a soul. For example, setting the motors to backwards when an object was in front of it became a way to play with the feeling of fear. Likewise, a fast forward motion when an object was in front gave a sense of an intentional aggressive attack.

Mediated Pulse

Physical Form

The piece is a composite prototype, combining hand blown glass with behavioral complexity. It is an exploratory sketch (Buxton, 2007) to research the combination of computational elements with a three-dimensional shape. The shape and the computational form are designed to give associations to an abstract, organ-like object (see Figure 5).

Technically Mediated Pulse consists of a microcontroller, battery, a vibration motor and an individually addressable *neopixel* light RGB string. The WIFI module in the microcontroller enables it to connect wirelessly to other inputs such as a pulse sensor. Furthermore, the wireless connection and battery allow the heart to be passed around in an audience context without being constrained by cables.

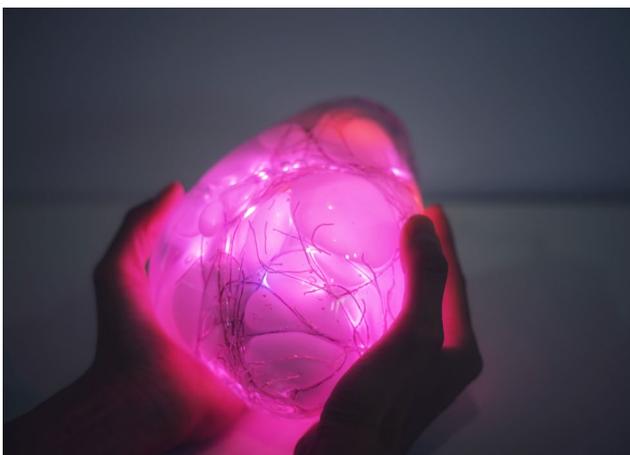


Figure 5. The organ-like glass heart which can vibrate and light up.

Behavioral Complexity

The behavioral complexity consists of expressing a sensed pulse of a dancer with a vibration motor and an animation on a LED string. The simple mediation of a dancer’s pulse can hardly be considered non-trivial complexity. Thus, the design example is somewhat an outlier to the core ideas of behavioral complexity presented in this paper. Its purpose is mainly to illustrate that the physical form, the organ-like shape combined with the pulse animation, creates expressive complexity with minimal non-trivial complexity. Two behavioral complexity strategies have somewhat been used: Multiple Layers and Alive Connotations (see Table 3).

Expressive Complexity

Mediated Pulse is intended as a platform to explore the potential of interactive audience experience. It has been used to visualize the pulse of a ballet dancer for the audience in a performance at the Royal Danish Theatre (see Figure 6). By attaching a wireless pulse sensor to the ballet dancer, it is possible to produce a representation of the pulse in the glass heart. This is done by making the heart vibrate in sync with the heartbeat and producing a synchronised red expressive animation in the string of light to give associations of blood running through the veins.

N7331227

Physical Form

N7331227 is an old industrial toilet seat grinder robot. Compared to the standards of modern robot technology, it is outdated in lacking



Figure 6. Ballet dancers exploring the Mediated Pulse. A pulse sensor is strapped to the arm of the dancer. The two seated dancers sense his pulse wirelessly through the heart-shaped object.

Table 3. Two behavioral complexity strategies have been used for Mediated Pulse.

Multiple Layers	Alive Connotations
The heart has both a light animation and a vibration pulse. Both adjust linearly to the dancer’s pulse, but with different animated timing sequences. The light animates as if red blood flows through the veins, while the motor spins up and down in sync with the pulse.	The combination of the physical form <i>a heart like shaped organ</i> and the pulsating light rhythm creates the expression of the object having an organ/heart aliveness.

the dynamic and flexible joints expected of a modern robot and the internal computer was only able to navigate a set of fixed points. Age became apparent in its aesthetic appearance (see Figure 7).

Behavioral Complexity

The robot was modified with open-source microcontrollers to control the joints and two cameras were mounted. One on its nose for computer vision tracking of passers-by and one from above to get an overall sense of movement in the space. When people approached the robot, it would detect them in its vicinity and start to *look* at them. When the robot had locked onto a passer-by, it would visually follow them through the room for a little while. Multiple modes were programmed into the robot. For instance, the robot would idle around when nobody was there and interact when somebody approached it. These modes gave the robot multiple moods to perform in the exhibition space. Three behavioral complexity strategies have been used: Reactiveness, Multiple Modes, and Alive Connotations (see Table 4).

Expressive Complexity

N7331227 is the serial number of the old industrial robot used. The interest was to reanimate the robot with an ingrained personality, for example, through its jerky and squeaky movements and an interest in creating an emotional relationship with people. Academically, N7331227 has been presented as a way to discuss the potential of animating non-living objects and how participants are able to create meaning around this (Hoby, 2014).

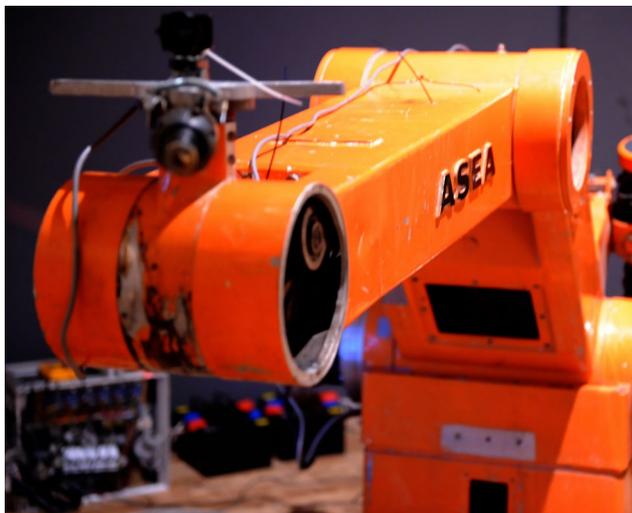


Figure 7. The industrial robot with a computer vision camera mounted on top to detect participants as they move closer.

Singing Plant

Physical Form

The Singing Plant is an interactive sound and light installation using a living greenhouse plant as the sole interactive interface element (see Figure 8). It is based upon one of the first electronic musical instruments, the Theremin, named after its inventor, the Russian professor Léon Theremin.

Behavioral Complexity

The Theremin works by sending an AC signal to an antenna and measuring the attenuation and distortion of the signal by the watery capacitance of a human body nearby.

Normally, the antenna is metal, but in the Singing Plant, a plant is used as the antenna. The water in the plant conducts well enough to make this possible, however great care in calibration is required as the electrical characteristics of the plant and its soil change with varying wetness. When properly calibrated, the Theremin-plant acts as a touch and proximity sensor, which controls pitch and volume. When the plant is touched, it gives feedback in the form of sound and light. The more participants touch it, the more energetically it responds. The sound is modulated through several filters to give a richer and more variable soundscape. Three behavioral complexity strategies have been used: Reactiveness, Multiple Layers, and Non-linearity (see Table 5).



Figure 8. Participants touching the Singing Plant to explore its soundscape.

Table 4. Three behavioral complexity strategies have been used for N7331227.

Reactiveness	Multiple Modes	Alive Connotations
The robots detect and tracks participant's faces in real time as they move around in the space. This creates a reactive system where the bodily movement of the participants produces a reaction in the movement of the robot.	The robot had five modes it could choose to use depending on the contextual situations. Idling around and looking for people, track people around in the space, look for a new drawing, interact with a light panel.	Many computational elements have been introduced for the robot to appear alive. Most prominently is the face detection and tracking system, which gives the impression of the robot being aware of people around it.

Table 5. Three behavioral complexity strategies have been used for the Singing Plant.

Reactiveness	Multiple Layers	Non-linearity
There is a direct feedback loop between touching the plant and getting a sound response. This allows the participants to explore different ways of holding and touching.	The plant has multiple layers of accumulative weighting factors that adjust based on the interactions. With prolonged interactivity, the light in the room dims and a pinspot shines the light on the plant only.	To create a space for exploration beyond a simple binary <i>touch equals sound</i> reaction pattern, multiple non-linear algorithms have been created to detect things like activity and amount of touch over a prolonged time. Further, multiple sound filters with similar properties have been applied: chorus, echo, flanger, etc.

Expressive Complexity

The Singing Plant has been discussed as a way to exemplify different touch interfaces for social play and exploration (Hoby, 2014; Hoby et al., 2013), highlighting the novelty of adding an interactive light and soundscape to an organic object.

Annotations: Strategies for Behavioral Complexity

We suggest five annotated strategies for behavioral complexity as computational form. Through the examples presented in the portfolio, we identify a set of prevailing strands to enable other designers to have an informed discussion about the overall potential.

To honor the ideal of computational form as a generative knowledge contribution, we have included code patterns for each strategy. They are examples of behavioral complexity in practice. Most of them have been extracted from the examples in the portfolio. Here they have been stripped from larger dependencies. Instead, the common Arduino (see homepage of Arduino website: <https://www.arduino.cc/>) syntax concepts like `digitalWrite` and `analogRead` are used to exemplify input and output.

Create Reactiveness

The concept of expressive complexity is somewhat misleading when reflecting on the potential of expressions in interactive systems. The system does not only express, but also react to the surroundings. The expressions are a whole between the input, the outputs and the behavioral complexity of the system. Based on its internal logic, it constantly reacts to the inputs it gets from its surroundings. A reaction pattern can simply be summarized as seen in Source code 1.

The case is here to present a simple action-reaction scenario. The simplicity of this case is such that it can hardly be considered complex in its behavior. Reactiveness prevails throughout most of the following strategies and thus serves as a basis for behavioral complexity.

Figure 9 shows the essential computational form as the feedback loop between input and output with behavioral complexity as the mediator. Vallgård (2014) express a similar term: computed causality, where the computer is used as the controlling property “to create the link between a cause-event and an effect-event” (p. 583). Through computational control, the causality “can be moderated, exaggerated, or entirely made up” (p. 583).

Playing with real-time reaction as an integrated part of the expression is a fundamental principle in most of the pieces in the portfolio. It is our observation through the experiments that

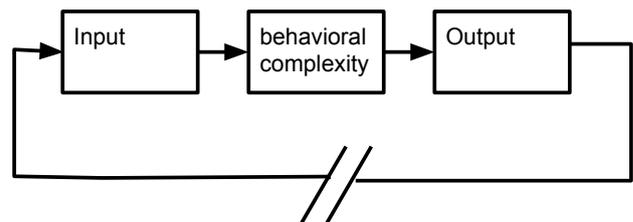


Figure 9. Boiled down to its essence, computational form within interaction design becomes a feedback loop between input and output (in the physical form) with the behavioral complexity as the mediator.

```
int buttonPin = 4;
int ledPin = 3;

void loop()
{
    // Turn on an LED if a button has been pressed
    if (digitalRead(buttonPin) == HIGH)
    {
        digitalWrite(ledPin, HIGH);
    }
    else
    {
        digitalWrite(ledPin, LOW);
    }
}
```

Source code 1. Pattern: A simple reactive system. In this case, an input (e.g., a button is pressed) results in some output (e.g., a light turns on). The reaction is instant and gives the impression of a clear connection between the interaction with the system and the reaction from the system.

real-time reaction holds an important dimension in understanding the expressions of the designed objects. Converting touch to sound in the Singing Plant created an artificial sense of aliveness. The slightest movement of the hand would change the pitch of the sound, thus giving the audience a sense of relating bodily to the plant. If the reactions to the input become too random or complex there will be a perceived loss of understanding of the relationship between the interaction, internal logic and the expressions of the system. For example, in the case of the Singing Plant, it became difficult for each individual participant to discern their own interaction when multiple people were touching it because the plant reacted to the accumulated amount of touch.

In scenarios that deviate from real-time feedback, it is at the cost of the possibility for the participant to interactively decode the expression. This is the case of the Mediated Pulse where the connection between the dancer's heartbeat and the pulsing heart is more conceptual. A participant holding the heart would not be able to discern the source of the pulse by interacting with the heart.

Create Multiple Modes

One recurring strategy for creating behavioral complexity is to have multiple modes of expression. Both Hoby (2014) and Larsen (2015) elaborate on this property with similar types of modes. Larsen describes three different modes of behavior: when something is sensed, idle state behavior when no interaction is present and wide sensing behavior when being aware of surroundings.

Typically, these modes have a binary threshold. Internal logic will decide which mode is the best fit for the current situation and act accordingly to the logic embedded in the mode itself. One such example was designed in a student project (see Figure 10) with the Animism Robot. The robot had an internal logic that would decide to be aggressive if something blocked its way a certain number of times. The internal mode would then shift from object avoidance to aggressive forward pushing, symbolizing an animal that had lost its patience.

Within computer science, a state machine is the simplest example of implementing a logical threshold for multiple modes. Each state has a set of code components that will be executed whenever the state is activated. In Source code 2 it is implemented with a switch statement.

Both the Animism Robot and N7331227 had multiple modes of behavior. N7331227 had idle mode when no presence was detected. Here it would look around to see if somebody were hiding in the corners of the room. In the following mode, it would keep an eye on participants while they moved through the space. In interactive mode, it would focus on a participant's face. There is no guarantee that two participants interacting similarly with the same system will receive the same expressions from the system because the internal mode may have shifted between the two interactions.

The level of behavioral complexity in this strategy becomes a matter of two things. Firstly, how complex is the individual complexity for each mode and secondly, how the change of mode is designed. For instance, one can have multiple simple modes with complex transition logic. Vice versa, one can have simple transitions between modes with a high level of behavioral complexity. In the second example, in the modes, it would be natural to use some of the other strategies presented.

Create Non-linearity

A noble aim within interaction design is to create some form of linear correlation between output and input. When you turn the light dimmer up, the light gets brighter and when you turn it the other way it dims. It makes sense in the sense of mimicking ways of navigating the natural world intuitively. However, as Vallgård and Sokoler (2010) phrase it: "The computer's ability to compute based on an input and to make the result available through an output



Figure 10. A customized version of the Animism Robot with embedded Anthropomorphic behavior.

```

unsigned long lastActivity = 0;
int buttonPin = 2;
int mood = 0;

void loop()
{
  // Detect some activity
  if (digitalRead(buttonPin) == HIGH)
  {
    lastActivity = millis();
  }

  switch (mood) {
    case 0: // alive mood
      // Jump to idle mood when no activity is present
      if (millis() - lastActivity > 10000)
      {
        mood = 1;
      }
      break;
    case 1: // idle mood
      // Jump to a live mode when activity is present
      if (millis() - lastActivity < 10)
      {
        mood = 0;
      }
      break;
  }
}

```

Source code 2. Pattern: A simple mode or mood changing system. In this example two modes are present. One mode in which the interactive system is "active" and one mode in which the system is "idle". When nobody has pressed a button for 10 seconds then the system goes in idle mode. This can be used to create the expression that the system is being bored when not interacted with.

means that in principle it can establish any desired cause-and-effect. The computer can thereby be a powerful tool in playing with our experience of the laws of nature" (p. 8). Similarly, Reeves, Benford, O'Malley, and Fraser (2005) argue "[t]he use of non-linear mappings to partially obscure the relationship between manipulations and effects is common in artistic installations where it introduces a degree of ambiguity in an attempt to provoke curiosity" (p. 745).

With Electrolumen, great effort was taken to expand the potential interaction space. Instead of just considering a touch connection between two poles as a binary decision (connection or not connection), we created three types of non-linear touch interpretations (see Figure 11): 1. the amount of touch, 2. amount of change in the amount of touch and 3. calculation of activity over time, based on the amount of change over a longer period of time. The first is the actual input with the two others being mathematically derived from the first. Source code 3 is an example of what the code could look like for a system with multiple non-linear parameters.

```
float raw = 0;
float amountOfTouch = 0;
float amountOfTouchOld = 0;
float amountOfActivity = 0;
float amountOfChange = 0;

void loop()
{
  raw = analogRead(A0);
  amountOfTouch = amountOfTouch * 0.9 + raw * 0.1;
  amountOfChange = amountOfChange * 0.9 + abs(amountOfTouch -
  amountOfTouchOld) * 0.1;
  amountOfActivity = amountOfActivity * 0.99 + amountOfChange *
  0.01;
  amountOfTouchOld = amountOfTouch;
}
```

Source code 3. Pattern: A non-linear system with multiple parameters. Above, a non-linear system is presented. It takes an input (raw) as an analog variable, filters it of noise and stores it in the variable amountOfTouch. The variable is used to calculate multiple non-linear parameters. For example, the amount of change is derived.

Although the second and third value did not have a linear relationship to the input, they very much correlated to the input and thus still provided a basis for expressions that were reactive to the interactions of the system. Although all three varied in the directness of correlation to the interaction, all of them had some level of symbolic link to the interaction itself. For example, the non-linear strategy of change connected with the sense of tapping. In the case of the Electrolumen, the three values were primarily presented through sound, affecting elements like pitch, volume and modulation. This combination became a multi-layered approach to creating expression, which we explain in the following strategy.

Create Multiple Layers

Where a non-linear strategy focuses on possible ways of interpreting the input, this strategy focuses on ways in which the different non-linear elements can be combined into a multi-layered expression.

In the Multiple Modes strategy, switching between different modes was based on a binary threshold. This meant that it was not possible for the expression to be a part of two moods or modes at the same time. Either a design can be bored or happy. It cannot not be bored and happy at the same time. One way to bypass this binary logic is through accumulation (Hobye & Löwgren, 2011; Vallgård, 2014). Vallgård writes that by accumulating over time “one state of expression becomes gradually more explicit than the other state” (p. 583). Source code 4 shows a simple gradual change between two states.

Singing Plant used accumulation to make a transition between the two states: idle and active mode. This is similar to the multiple modes in the second strategy, but instead of a binary transition between the modes, the plant would gradually move between the two. If somebody came and interacted with the

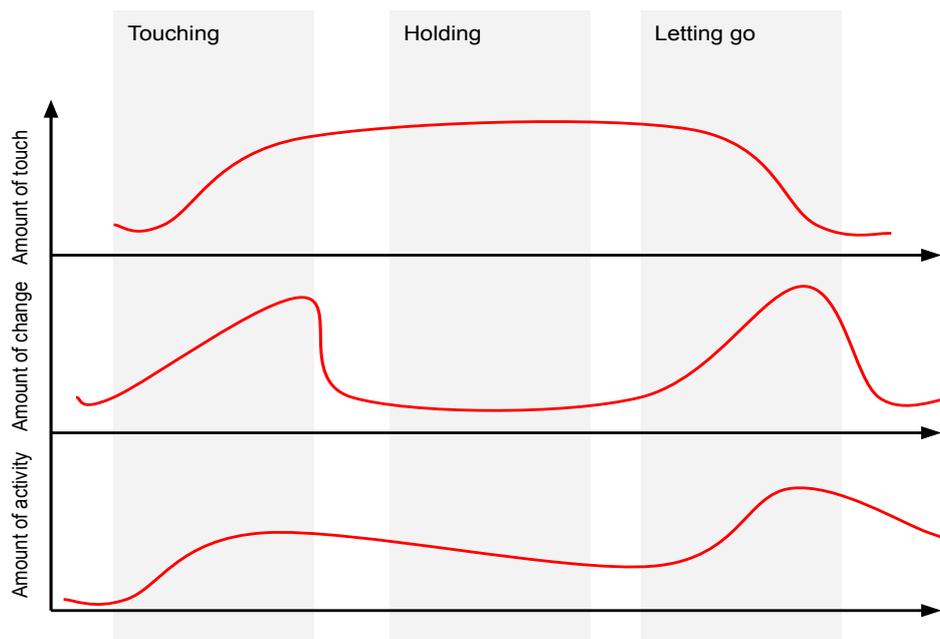


Figure 11. An illustrative example of the internal non-linear system of the Electrolumen (Figure from Hobye, 2014, p. 196).

```

int activityFader = 0;
int amountOfTouch = 0;
int redLedPin = 10;
int blueLedPin = 11;

void loop()
{
  amountOfTouch = analogRead(A0);
  if (amountOfTouch > 10) {
    activityFader = activityFader + 1;
  } else {
    activityFader = activityFader - 1;
  }
  activityFader = constrain(activityFader, 0, 100);

  analogWrite(redLedPin, map(100 - activityFader, 0, 100, 255, 0));
  analogWrite(blueLedPin, map(activityFader, 0, 100, 0, 255));
  delay(100);
}

```

Source code 4. Pattern: A crossfader with multiple layers. The pattern fades between two LEDs (red and blue) over time when a sensor has been touched. The red light will fade up when the blue light fades down and the other way around.

plant, the light in the room would slowly dim while the spotlight directly on the plant would brighten. If there were a pause in the activity, the light would naturally gravitate towards the idle mode of light in the room and there would be no light on the plant. This meant that the transition between the two modes was relative to the activity level and thus would shimmer back and forth as the amount of activity changed.

By contrast, accumulation becomes a gradual transition between two states, a multi-layered strategy that can be extended to create real-time space for exploration. By combining multiple parameters, it is possible to create behavioral complexity that gives a sense of multi-layered expressions. It does not really have specific states, but instead certain elements can come alive in different ways through the interaction. If enough layers are added, it becomes hard for the designer to be able to predict the different combinations (see Padfield & Andreasen, 2012).

Electrolumen used this extended strategy. It mixed multiple non-linear interpretations of touch (see Figure 11). By combining multiple non-linear interpretations of touch, a more variable soundscape was created for the participants to explore; tapping would modulate the soundscape in a different way than statically holding, etc. Combining Multiple Layers with Non-linearity properties, we argue for the potential of considering the different layers as *interweaved multidimensionality*, which unfolds with the interaction.

Create Alive Connotations

In many ways, behavioral complexity revolves around how to express liveness in non-living computational objects. Create Alive Connotations is when we play with the connotations of resembling a living thing (e.g., animal, human, plant, or organ)

primarily through computational patterns, but as a strategy can play with both behavioral complexity and physical form for expressive complexity. In relation to behavioral complexity, the concept of animism (Larsen, 2015) comes closest to describing the potential of having behavioral complexity for expressive complexity. Animism is concerned with the potential of creating alive connotations in things that do not necessarily have human or animal form. The entanglement between behavioral complexity and expressive complexity calls for some reflection. In our initial distinction, we pointed out that other factors could contribute to the actual expression beyond mere code. Sometimes, the actual behavioral complexity is somewhat simple, but combined with physical properties it becomes a rather powerful expression.

Because of the Animism Robot's zoomorphic visual appearance, a minimal amount of code is needed for it to become *alive* in its expression. The infrared sensor resembles eyes and the configuration of the servos mounted on top of the robot platform gives association to a neck and a head. Therefore, adding code that tilts the head up/down and right/left is enough to give the impression of an emotional expression of the robot. A similar anthropomorphic property occurred in N7331227. What objectively speaking was a mechanical robot arm with a camera mounted on top became alive as if the arm itself was a body and the camera an eye. Even though basic movement of the robot arm would create alive connotations, a much more vivid expression was present when the robot arm used the camera to track participants in the space. It would create a real-time connection with the bodily movement of the audience. If they shifted to the left, the robot's head would follow suit and so forth. The robot's *awareness* of the participant's presence in the space gave a sense of it coming alive as if it had its own agency. The Animism Robot has also been used to illustrate this example. Programming the robot to move backwards if somebody were *too close* created the expression of wanting to maintain a safe distance towards other people in the space. Source code 5 shows the code for such example.

Turning touch into sound in the singing plant gave added connotations of aliveness. Where the plant itself was organic, a similar expression appeared with Mediated Pulse. The shaped form of a heart gave a sense of an organ-like expression, while the light pattern and the vibration gave it a sense of alive connotations. It was common for participants to react with sparkling eyes (Hoby & Löwgren, 2011) whenever they picked up the heart (see Figure 12).

Mediated Pulse is the least complex design in the portfolio from a behavioral complexity point of view. It vibrates and makes light animations based on the human pulse. However, it is relevant to include because the power of creating a *rhythm* (Vallgård, 2014) in this case gave a sense of alive connotations even though the code was not complex. One such example is seen in Source code 6.

Both code examples in this strategy are surprisingly simple compared to the vivid expression they produced in combination with the physical form. They show that expressive complexity is a product of a whole and not just the behavioral complexity in the computational form.

```

int distance = 0;

Servo leftWheel;
Servo rightWheel;

void setup() {
  leftWheel.attach(9); // attaches the servo on pin 9 to the servo object
  rightWheel.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  distance = analogRead(A0);

  // do something when distance sensor read a value below 300
  if (distance < 300)
  {
    // Make the continuous servo motors move backwards
    leftWheel.write(70);
    rightWheel.write(110);
  }
  else
  {
    // Stop the continuous servo motors
    leftWheel.write(90);
    rightWheel.write(90);
  }
}

```

Source code 5. Pattern: *Creating alive connotations through distance reaction.* This pattern maintains a certain distance to its surroundings. If a person is too close the robot will back off until the distance criteria are met.



Figure 12. The ballet dancer senses her own heartbeat through the Mediated Pulse.

```

int ledPin = 1;
unsigned long timer = 0;
boolean on = false;

void loop()
{
  if (millis() - timer > 2000) // do something every two seconds
  {
    timer = millis();
    on = !on;
    digitalWrite(ledPin, on);
  }
}

```

Source code 6. Pattern: *Create rhythm/pulse through timing.* By timing an interval this example can turn on and off, e.g., a LED, creating a pulse or rhythm that can give associations to a heartbeat or breathing.

Discussion: *Unpacking Implementation Strategies*

We have now presented five annotations to illustrate the concept of behavioral complexity as a computational design strategy to create expressive complexity. The five annotations set the stage for us to reflect on possible practical implementations. Although the strategies alone may elicit some level of participant exploration, how they are combined and how they play into the overall expression through the physical form is crucial for us to understand the potential of the overreaching design strategy. Therefore, in the following, we revisit our portfolio to discuss the nuances in the implementation of the strategies in practice. We look into the possibility of combining multiple strategies, strategies for preserving coupling and how the expressive complexity is a product of both the behavioral complexity and the physical form.

Combining Strategies

All of the designs in the portfolio make use of multiple behavioral complexity strategies. Table 6 maps out the different design examples in relation to the behavioral complexity strategies.

When mapping out the different expressions a few patterns start to occur. First and foremost, all of the design examples use 2-3 strategies indicating the need for at least two. It also seems that three is enough. Further, it can be seen that Reactiveness is a recurring strategy throughout the portfolio. This makes intuitive sense since it provides the participant with the possibility to iteratively explore different interactions to understand the possibility space. Mediated Pulse is the only one without Reactiveness. From the perspective of participant interaction, the behavioral complexity is a conceptual understanding of the connection to the performer's pulse. If the participant interacting with the object was also wearing a pulse sensor it may create a fundamentally different level of expressive complexity. With the version presented in the portfolio, the alive connotations

Table 6. The mapping between behavioral strategies and the design examples in the portfolio.

	Reactiveness	Multiple Modes	Non-linearity	Multiple Layers	Alive Connotations
Electrolumen	•		•	•	
Singing Plant	•		•	•	
N7331227	•	•			•
Animism Robot	•	•			•
Mediated Pulse			•		•

mostly lie in the physical form of resembling an organ and how the different animation patterns support this notion. The role of the Mediated Pulse therefore also shows a deviation of the core interest in the portfolio.

Two more patterns are immediately apparent. The last four projects in the portfolio can be grouped into two groups. Electrolumen and Singing plant share Non-linearity and Multiple Layers. N7331227 and Animism Robot both use Multiple Modes and Alive Connotations. N7331227 and Animism Robot both share the same interest in creating human-like traits. The multiple modes create the expression of an ability to change personality (e.g., from bored to happy) and the alive connotations creates a reaction pattern of them relating to the real world. With Electrolumen and Singing Plant their physical expression differs greatly. The plant is a natural organic living organism and Electrolumen is a rough industrial light pole. It is the only design in which the physical form does not play with the connotation of resembling a living thing. The reason for them to share the same two strategies is that they both explore the potential of creating analog reaction patterns of touch. The Non-linearity generates complex touch patterns for a multilayered expression.

Tieben et al. (2011) argue for the complexity of the system as one of multiple strategies to prolong the discovery of an installation. In the portfolio the general motivation behind combining behavioral complexity strategies is to prolong the time that it took for participants to decode the possible patterns in the system.

Preserving Coupling in Complexity

Designing with behavioral complexity can come at the cost of losing coupling for the participant interacting. For a design to be deemed interactive, it is necessary for the participant to have some level of understanding of how their interaction affects the system. This becomes a paradox. At some level one is interested in creating interesting and complex interactions, but on another level, interactions should not be so complex that the participant does not understand the coupling. Gaver et al. (2009), Hoby (2014) and Larsen (2015) all consider this paradox. Gaver et al. (2009) express how the lack of *sweet spot* between randomness and accuracy in a system can affect the interpretation: “The outputs were seen as wrong quite often, to the extent that at least some participants speculated that the sensors might simply be fakes” (p. 2215). Svanæs (2013) adds an embodied phenomenological point to coupling that “the action-reaction

coupling should be one that is easily ‘understood’ by the body” (p. 26). Hence, “we should consider interaction techniques that allow for rapid coupling between user actions and system feedback” (p. 26). In our definition of behavioral complexity, we highlight this paradox by both requiring an action/reaction relationship of the environment and by pointing towards the design of non-linear and non-deterministic properties.

The need for preserving coupling may be one reason why none of the design examples deploys more than three types of strategies at the same time. Further, the risk of the participant losing coupling is more present in some strategies than others. Reactiveness and Alive Connotations do not inherently come with the cost of loss of coupling because both seek to create vivid real-time interaction with the participants. Whether the strategy of creating Multiple Modes affects coupling greatly depends on the different kinds of modes deployed and the logic behind the choice of mode. If the modes are connected meaningfully to the interaction, the participant will not lose the sense of what is going on. Conversely, if the robot changes the mood in an unpredictable way, the participant will lose their own coupling to the interaction.

The strategies Multiple Layers and Non-Linearity both deliberately push the boundaries of coupling. They seek to open the interaction space to complex interaction patterns that need exploration and allow higher levels of mastery by the participant. In the design examples from the portfolio, a strategy of dividing different non-linear layers between different mediums has been used. With the Singing Plant, the lights gave an overall sense of the accumulation of energy, while the sound output was more directed to touch interactions. Electrolumen used light for direct and clear feedback of touch and a complex soundscape in which multiple parameters affected the sound output (see Figure 13). By dividing the different parameters into different physical expressions, it is possible to give a sense of behavioral complexity without compromising a sense of direct coupling to the interaction itself.

Preserving coupling in complexity can be seen as a matter of catering for both masters and novices at the same time. Clearly coupled feedback, for example, in the form of light, allows the novice to grasp the basic interaction. When the initial coupling is mastered, the fine nuances embedded in the behavioral complexity allow for new interactions. By separating coupling and complexity into two different mediums, both can exist at the same time. Alternatively, one could use an accumulation strategy to dynamically adjust the amount of behavioral complexity based on an assessment of the participant’s skill level.

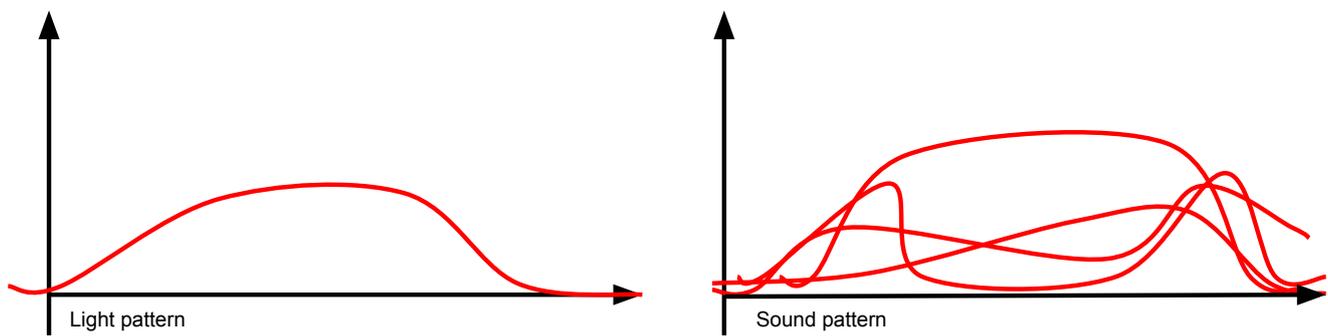


Figure 13. Illustration of using the different non-linear patterns as an expression on different mediums. The light has a direct coupling to touch and the sound have a more complex mix between multiple parameters (Figure from Hoby, 2014, p. 101).

Conclusion

Multiple academic voices have similar interests in discussing behavioral complexity, although at first glance they may seem to have varied agendas when it comes to the experiential qualities they intend to elicit. In the greater picture, concepts like curiosity (Tieben et al., 2011), play (Gaver, 2009), novelty (Gaver et al., 2003; Hoby, 2014), extended material qualities (Vallgård, 2014), ambiguity of information (Gaver et al., 2003) and animism (Larsen, 2015) are close cousins when it comes to behavioral complexity to create alive and adaptive expressions.

Our knowledge contribution follows our threefold intention with the paper. Firstly, we have argued for a shared agenda within interaction design for a further exploration of computational complexity within computational material as a resource in design. We have illustrated this through a model that introduces the concept of behavioral complexity as a computational design strategy to create expressive complexity in order to propose that computational complexity is a central part of a form giving practice within interaction design.

Secondly, we have exemplified the relationship between behavioral complexity and expressive complexity through five design examples in an annotated portfolio where we suggest five strategies of behavioral complexity exemplified with programming patterns to serve as inspirational building blocks for designers/developers: create Reactiveness, create Multiple Modes, create Non-linearity, create Multiple Layers and create Alive Connotations.

Thirdly, we have initiated a discussion of the pros and cons of combining the strategies and how the mapping of different strategies in combination affects expressive complexity. We conclude that there is a need for balancing the sweet spot and preserving coupling in complexity.

We do not consider our design strategies to be an all-encompassing taxonomy. Instead, we see the properties and strategies as a practice to be further evolved. For example, with the adaptation of technologies like artificial intelligence and machine learning, one can expect the system to behave in more complex ways than we can yet imagine.

Acknowledgements

The Singing Plant

Mads Hoby, Nicolas Padfield, Schack Lindemann, Thomas Jørgensen, Thor Lentz, and Åsmund Boye Kverneland.

Mediated Pulse

Mads Hoby, Maja Fagerberg Ranten, and Nina Gram. Done in collaboration with the Royal Danish Theatre. Glass form development as a part of Dynamic Transparencies with Henrik Svarrer Larsen, Peter Kuchinke and Mads Hoby. Blown by Bjørn Friborg.

Animism Robot

Mads Hoby, Nicolas Padfield, and Nikolaj Møbius. Done as a part of Fablab RUC.

N7331227

Brian Josefsen, Eva Kanstrup, Jonas Jongejan, Mads Hoby, Nicolas Padfield, Nikolaj Møbius, Schack Lindemann, Thomas Fabian Eder, and Thomas Scherrer Tangen. Done as a part of illutron.

Electrolumen

Done as a part of a large-scale installation by illutron with: Bent Haugland, Brian Josefsen, Christian Liljedahl, Emma Cecilia Ajanki, Halfdan Jensen, Jacob Viuff, Jiazi Liu, Johan Bichel Lindegaard, Jonas Jongejan, Lasse Skov, Lin Routh Jørgensen, Lizette Bryrup, Mads Hoby, Marie Viuff, Mathias Vejerslev, Mona Jensen, Nicolas Padfield, Nikolaj Møbius, Philip Jun Kamata, Sally Jensen Ingvorsen, Schack Lindemann, Simo Ekholm, Sofie Kai Nielsen, Sofie Walbom Kring, Sofus Walbom Kring, Sonny Windstrup, Tobias Bjerregaard, Tobias Jørgensen, Troels Christoffersen, and Vanessa Carpenter.

References

1. Buxton, W. (2007). *Sketching user experiences: Getting the design right and the right design*. Burlington, MA: Morgan Kaufmann.

2. Bødker, S. (2006). When second wave HCI meets third wave challenges. In *Proceedings of the 4th Nordic Conference on Human-Computer Interaction* (pp. 1-8). New York, NY: ACM.
3. Fallman, D. (2008). The interaction design research triangle of design practice, design studies, and design exploration. *Design Issues*, 24(3), 4-18.
4. Gaver, B., & Bowers, J. (2012). *Annotated portfolios*. *Interactions*, 19(4), 40-49.
5. Gaver, W. W., Beaver, J., & Benford, S. (2003). Ambiguity as a resource for design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 233-240). New York, NY: ACM.
6. Gaver, W., Bowers, J., Kerridge, T., Boucher, A., & Jarvis, N. (2009). Anatomy of a failure: How we knew when our design went wrong, and what we learned from it. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2213-2222). New York, NY: ACM.
7. Gaver, W. (2009). Designing for homo ludens, still. In T. Binder, J. Löwgren, & L. Malmberg (Eds.), *(Re)searching the digital Bauhaus* (pp. 163-178). London, UK: Springer.
8. Gaver, W. (2012). What should we expect from research through design? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 937-946). New York, NY: ACM.
9. Hallnäs, L., & Redström, J. (2002). From use to presence: On the expressions and aesthetics of everyday computational things. *Interactions*, 9(4), 11-12.
10. Hallnäs, L. (2011). On the foundations of interaction design aesthetics: Revisiting the notions of form and expression. *International Journal of Design*, 5(1), 73-84.
11. Hoby, M., & Löwgren, J. (2011). Touching a stranger: Designing for engaging experience in embodied interaction. *International Journal of Design*, 5(3), 31-48.
12. Hoby, M., Padfield, N., & Löwgren, J. (2013). Designing social play through interpersonal touch: An annotated portfolio. In *Proceedings of the Conference on Nordic Design Research* (pp. 366-369). Copenhagen, Denmark: KADK, School of Design.
13. Hoby, M. (2014). *Designing for homo explorans: Open social play in performative frames* (Doctoral dissertation). Malmö University, Malmö, Sweden.
14. Larsen, H. S. (2015). *Tangible participation—Engaging designs and design engagements in pedagogical praxes* (Doctoral dissertation). Lund University, Lund, Sweden.
15. Levillain, F., & Zibetti, E. (2017). Behavioral objects: The rise of the evocative machines. *Journal of Human-Robot Interaction*, 6(1), 4-24.
16. Löwgren, J. (2002). *The use qualities of digital designs*. Retrieved March 20, 2018, from <https://pdfs.semanticscholar.org/260e/8360b33b8ac003a5e244a13396471cd9a47a.pdf>
17. Löwgren, J. (2007). Forskning om digitala material [Interaction design, research practices and design research on the digital materials]. In S. Ilstedt Hjelm (Ed.), *Om design forskning* (pp. 150-163). Stockholm, Sweden: Raster Förlag. Retrieved from <http://jonas.lowgren.info/Material/idResearchEssay.pdf>
18. Padfield, N., & Andreassen, T. (2012). Framework for modelling multiple input complex aggregations for interactive installations. In *Proceedings of the 2nd International Conference on Ambient Computing, Applications, Services and Technologies* (pp. 79-85). Roskilde, Denmark: Roskilde university. Retrieved from <https://rucforsk.ruc.dk/ws/portalfiles/portal/38396286>
19. Padfield, N., Haldrup, M., & Hoby, M. (2014). *Empowering academia through modern fabrication practices*. Paper presented at Fablearn Europe, Aarhus, Denmark. Retrieved from <https://rucforsk.ruc.dk/ws/portalfiles/portal/49670>
20. Redström, J. (2017). *Making design theory*. Cambridge, MA: MIT Press.
21. Reeves, S., Benford, S., O'Malley, C., & Fraser, M. (2005). Designing the spectator experience. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 741-750). New York, NY: ACM.
22. Svanæs, D. (2013). Interaction design for and with the lived body: Some implications of Merleau-Ponty's phenomenology. *ACM Transactions on Computer-Human Interaction*, 20(1), Article 8.
23. Tieben, R., Bekker, T., & Schouten, B. (2011). Curiosity and interaction: Making people curious through interactive systems. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction* (pp. 361-370). Swinton, UK: British Computer Society.
24. Vallgård, A. (2014). Giving form to computational things: Developing a practice of interaction design. *Personal and Ubiquitous Computing*, 18(3), 577-592.
25. Vallgård, A., & Sokoler, T. (2010). A material strategy: Exploring the material properties of computers. *International Journal of Design*, 4(3), 1-14.
26. Vallgård, A., Winther, M., Mørch, N., & Vizer, E. E. (2015). Temporal form in interaction design. *International Journal of Design*, 9(3), 1-15.

