What a Life! Building a Framework for Constructive Assemblies

Joanne Leong[1], Florian Perteneder[1], Hans-Christian Jetter[2], Michael Haller[1]
[1] Media Interaction Lab, University of Applied Sciences Upper Austria
[2] University of Applied Sciences Upper Austria

ABSTRACT
Constructive assemblies are tangible user interfaces (TUIs) that involve the interconnection of modular parts. As such, they offer a unique means to support us in our various and diverse activities. However, little information is available for understanding the intricacies of taking a modular, constructive assembly approach to TUI design. Based on an analysis of extensive data collected from interviews with eight world-class TUI experts, we propose a descriptive, conceptual framework to facilitate systematic investigation and critical consideration of constructive assemblies. The paper presents a lifecycle model for constructive assemblies and discusses their main design qualities and associated parameters. We demonstrate how this can be used to structure critical discussions by applying the principles to existing works and the design of our own constructive assembly.

Author Keywords
Tangible Interface; Tangible Interaction; Framework; Design; Constructive Assembly

ACM Classification Keywords
H.5.2. User Interfaces

INTRODUCTION
Constructive assemblies are tangible user interfaces (TUIs) that involve the interconnection of modular physical, interactive *units* to formulate larger constructions that are automatically or manually put-together [7,17,28]. The modularity of such TUIs inherently allows for varying degrees of versatility in how they are used. As such, constructive assemblies have appeared and been leveraged in many domains ranging from storytelling and education to rapid prototyping and the manipulation of digital content, as pictured in Figure 1. However, despite the wealth of examples of successful constructive assemblies, we observe that there is a lack of guidelines for this topic which can serve to aid designers in systematically investigating constructive assemblies and creating new ones.

TEI '17, March 20 - 23, 2017, Yokohama, Japan

ACM 978-1-4503-4676-4/17/03…$15.00
DOI: http://dx.doi.org/10.1145/3024969.3024985

Figure 1. Examples of constructive assemblies: Tern [15] **a tangible programming exhibit at the Boston Museum of Science (left); Tiles that Talk** [6] **a system to aid students in prototyping interactive, networked systems; GaussBricks** [21] **a system of magnetic building blocks for use on portable displays (right).**

This absence motivated us to speak with experts in the field to elicit key thoughts and design cornerstones for constructive assemblies. In this paper, we present our findings from an in-depth analysis of interview data gathered through an extensive expert study. Summarizing, the main contributions of this paper are as follows:

- A *discussion of the benefits and risks* of a constructive assembly approach to TUI design
- A *lifecycle model* that outlines different phases that constructive assemblies can transition between, and how these phases are interconnected
- An outline of *central design themes* for constructive assemblies and a discussion of *parameters* that can be manipulated to achieve a desired balance of these qualities
- Three *case studies* based on examples from literature and our own design example, which demonstrate how these principles can be applied to elicit critical design thinking and discussion and can be used to shape future design iterations.

RELATED WORK

State of the Art for Constructive Assemblies
The category of TUIs referred to as *Constructive Assemblies* was explicitly identified by Ullmer and Ishii in 2000 [34], and was later reiterated as being one of seven promising directions for the development of new TUIs [17]. They are typically defined as TUIs which consist of interactive units that can be either manually or automatically connected to form more complex structures [7,17,28].

Constructive assemblies have been created for a variety of use cases. They were originally created for the purposes of modelling and simulation [1,2,4,9,10], but have since been leveraged for numerous other domains. For instance, constructive assemblies have been heavily used for hands-on education

[23,25–27,30,33,35,37], storytelling and other forms of entertainment [3,5,12,13], rapid prototyping [6,20,38], interactive public exhibits [15], and input devices for working with digital displays and content [11,21,36]. Furthermore, advancements in fabrication and actuation have spurred efforts to explore the boundaries of what is technologically feasible for this category of TUIs, and has thus been a central topic for recent works such as Changibles and Kinetic Blocks [28,29]. In research, a constructive assembly consisting of identical units has also been used as a medium with which to explore the concept of emergent interfaces and its associated parameters [7]. However, while there are numerous successful examples of constructive assembly TUIs, we observe that there is a lack of conceptual description, formalization and theory to guide their design. In this work, our aim is to address this gap by presenting a framework for such TUIs.

Positioning our Framework

Many conceptual frameworks for TUIs have been established, which differ in their focus and conceptual origin [24,31]. Some prominent examples include the Model-View-Controller pattern presented by Ullmer and Ishii [34], the containers, tokens and tools concept from Holmquist et al. [14], the tokens and constraints concept within the TAC paradigm introduced by Shaer et al. [32], and the 2D taxonomy of embodiment and metaphor by Fishkin [8]. Frameworks have also been established to understand TUIs within the scope of greater contexts and domains (e.g. within a social context [16], within the context of learning [22,37], and within the scope of reality-based interaction [18]). Given the large number of frameworks that have been created for TUIs, it can be challenging both to understand what a framework offers with respect to others, and to isolate a framework that best serves one's needs.

Fortunately, Mazalek and Van Den Hoven [24] created a map of TUI frameworks that brings frameworks into one conceptual space and helps readers to navigate between them. We use their definition of framework (i.e. a skeletal structure that can contain guidelines and can be used by designers to select a course of action in different stages of their design processes) and use their map to situate our work with respect to others. We identify our framework as being of the ***designing*** type, which covers the ***interactions*** and ***physicality*** facets of this space. This is because our work focuses on highlighting design considerations to guide the formation of new constructive assemblies, and aims to establish a better understanding of both user-interactions and physical affordances pertaining to constructive assemblies. In this work, we achieve this by conducting interviews with several TUI experts in order to gain insight on constructive assemblies. Later in this paper, we ground our framework by discussing our concepts in the context of examples from literature and our own design example.

STARTING POINT: CONSTRUCTIVE ASSEMBLIES

Constructive assemblies present an interaction paradigm shaped by their transient and ever adapting physical forms. Looking at examples in literature, we hypothesized a basic perspective of constructive assemblies involving three stages, as illustrated in Figure 2. A *creation* phase marks the beginning of a lifecycle, where interactive components or *units* are put-together to form an interactive whole or *assembly*. The assembled whole then typically serves a purpose to the user in a *usage* phase. The *destruction* phase closes one lifecycle, where the assembly is then disassembled into its constituent parts. A cycle can be established by recycling the disassembled components into the design of a new assembled whole.

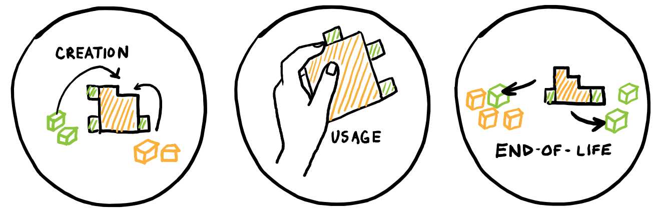

Figure 2. Basic interaction phases of constructive assemblies.

This initial concept was the starting point from which we decided to talk with experts to investigate and distill a more refined model of our interactions as users with constructive assemblies. We surmised that a refined model of the lifecycle would provide a framework for deeper and more structured critical discussion of constructive assemblies.

EXPERT STUDY

We talked to experts with the intention of eliciting key thoughts surrounding constructive assemblies and collecting responses that could ground our understanding of these TUIs.

Expert Study Participants

We consulted 8 experts (4 female) in the field of TUIs to conduct an expert study. The experts (E1-E8) interviewed came from both academic and industry backgrounds, and were based in world-renown facilities and institutions across North America and Europe. The group of experts with academic experience consisted of several leading researchers (professors and post-docs) who have extensive experience in the field of TUIs, and have on average 46 publications, many of which have been accepted to top-tier HCI conferences such as TEI, CHI and UIST. Those with industry experience work in large-scale international companies and have experience developing tangible interfaces that have been incorporated into products for end-users. Moreover, a few of the interviewed experts had experience in steering both academic and commercial endeavors to create TUIs for various contexts and scenarios.

Interview and Data Analysis Procedure

Each expert was asked a series of questions in a semi-structured interview process. The interviews were between one half to two hours in duration. Prior to the interviews, themes of interest and sample questions catered to the backgrounds of the experts were prepared. Interviewees were asked questions that fell into four general categories: *general design considerations*, *constructive assemblies*, *TUI special issues* and *user-involvement*. These questions were used as a loose structure for the interview process. Follow-up questions were asked to clarify and elicit more detailed explanations of the concepts that interviewees raised. Interviewees were encouraged to draw upon their past projects and experiences in creating TUIs as part of their responses. All interviews were transcribed and sectioned into quotes for a bottom-up, open coding approach to data analysis. Affinity diagrams were built, and groupings emerged that highlighted an array of problems, qualities, benefits and drawbacks, opportunities and risks, and approaches to creating constructive assemblies. The results of this qualitative analysis are explained in the following result sections.

RESULT 1: BENEFITS AND DRAWBACKS

All the experts that were consulted emphasized that the design of a TUI must always depend on the specific target problem and the goal you want to reach. In light of this, many experts pointed out a number of benefits and risks that can offer some assistance in determining whether a constructive assembly approach is suitable for a given problem.

There is a wide array of potential benefits for taking a constructive assembly approach to TUI design. Two experts were quick to express that the design space for constructive assemblies is, *"much wider"* (E6) and that they are *"versatile [and] endless in possibilities"* (E4). E4 added that they can be *"accessible to users"*; the notion of putting together pieces such as blocks is an experience many of us can relate to as children. Furthermore, positive implications for *performance*, *lifespan (*E1*)* and *attachment* (E6) were identified. One expert noted that a modular approach opens the doors to *optimization* and *customization*, leading to prolonged use of the TUI and increased *emotional attachment* (E1). Many experts expressed that constructive assemblies can be "*very beneficia*l" to learning (E1, E5, E7, E8); amongst the reasons listed were that they can provide "*enjoyment"*, and are "*more motivating [and] more engaging"* (E7). Two experts also added that the simple process of building meant users would develop an inherit understanding of the assembly, thereby circumventing the need to explicitly learn the tool (E1, E8). Another expert noted that they can *"encourage a kind of thinking [or] mindset"* (E6).

Experts cautioned that constructive assemblies could result in a *"very rectangular"* (E6) aesthetic that could be likened to the look of the video game Minecraft (E4). Particular technical concerns were also raised. E5 noted that having modular components could result in assemblies that are *"too bulky,"* or that may *"accidentally fall apart."* This expert also noted that the constant need to *"plug them together"* could in itself be a disadvantage to other types of TUIs. Furthermore, E8 pointed out that users may have limited capacities to effectively use the modules for construction, likening it to: *"giving them all the parts of a fishing rod, and they don't know what to do with it."*

RESULT 2: LIFECYCLE MODEL

In discussion with experts, it became clear that a more sophisticated model was needed to describe the lifecycle of constructive assemblies. Based on the interviews, we now augment our earlier hypothesized model of a three-phased lifecycle for constructive assemblies, which had consisted of a *creation phase*, *usage phase* and *destruction phase*. Figure 3 depicts the more complex and refined picture of the lifecycle that emerged from speaking with our TUI experts.

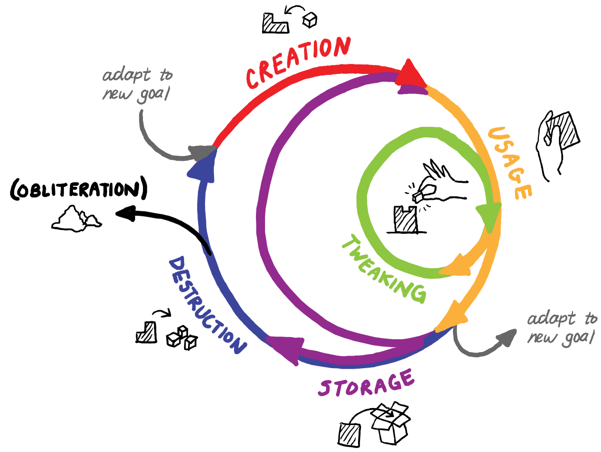

Figure 3. The constructive assembly lifecycle.

Lifecycle Phases

The refined model of a lifecycle for constructive assemblies consists of the following phases: ***creation, usage, tweaking, storage, destruction, and obliteration.*** A summary of the terminology is provided in Table 1.

In both the creation and usage phases, users can reflect on the design and assess whether the current state of the assembly is in alignment with their goal. Four of the experts referred to this concept with phrases such as *"trial and error"* (E5) and *"trying it out."* E2 explicitly pointed out a *"reflective phase."* In the proposed model, this concept is presented under both creation and usage. Users can evaluate the assembly in increments as they add additional pieces or evaluate an assembly as a whole after they believe their goal has been reached.

Conversations with experts resulted in three new phases being introduced to the lifecycle: a *tweaking phase*, a *storage phase*, and an *obliteration phase.* Three experts suggested that users interacting with constructive assemblies can *"enter a phase of rapid iterative testing and evaluation [and] adjustment,"* (E8) where the user is *"building on the fly,"* (E1) modifying the construction, and trying to *"improve it"* (E5)*.* E8 referred to this as a *"constant tweaking,"* inspiring the name for the ***tweaking phase***. Experts also referred to a stage where the constructed whole may simply be *"standing around"* because users may choose to "*not take them apart"* and *"keep it for a later specialized session of that type"* (E8). We refer to this as the ***storage phase***.

The possibility for the constructed wholes to be thrown away (E2) or *"disappear* or *break"* (E4) was also mentioned, suggesting that the components in the assembly enter a state where they can no longer be used, and can more extensively make the system, as an assembly, unusable. We refer to this phase then as the ***obliteration phase***.

Phase Transitions

Each arrow represents a phase in the constructive assembly lifecycle, where the phases *"entwine [and] flow into each other"* (E8). This blending of phases is visualized by the overlapping of certain arrows. Entry points and exit points for the

Table 1. Lifecycle phase descriptions.

Phase	Description
Creation	When a user has a goal in mind, and pieces together the modular components in incremental steps to achieve this goal. While building, users usually evaluate their construction.
Usage	The user interacts with the constructed whole, and evaluates it in the process.
Tweaking	The user adds, modifies, or removes components in order to improve the construction.
Storage	A constructed whole is kept, but not used.
Destruction	The phase in which users take-apart the assembly into its constituent parts.
Obliteration	The phase in which the pieces of the constructive assembly are no longer functional.

lifecycle are marked by gray arrows leading in and outwards of the circle. These correspond with users adapting their goals. For instance, a user's goal may transition from simply trying to game a system and find out its limitations (E1), to trying to get it to do *"what you want it to do"* (E8).

Closing the loop
The cycle is created with the overlapping of the destruction and creation phases. While it may be that the modular components are not ultimately re-appropriated for use in new constructions, many experts' comments referred to the possibility of repurposing and reusing modules (E1,E2,E3,E5,E8). Furthermore, two experts saw destruction and creation as tightly interlinked, where destruction was seen as a *"prerequisite for another construction phase"* (E1) and saw disassembly as being motivated by assembly (E8).

Value of Lifecycle Concept
Knowing and considering the opportunities and challenges that are highlighted when considering TUIs from the perspective of their lifecycle can help us focus our overall design approach. Directing emphasis towards a particular phase of the lifecycle can help designers achieve specific design goals. For instance, if the TUI should engage a user to explore a multitude of ideas in high-frequency design cycles, a crucial aspect to the design will be to ensure the system consists of components that can be easily taken apart and are interchangeable.

RESULT 3: DESIGN THEMES AND PARAMETERS
In conversations with experts, many different aspects of consideration for constructive assemblies were discussed. Through the coding of conversations followed by affinity diagramming, we found a number of themes to emerge from the interview data. In this section, we first outline these themes, and propose an overarching structure for how they relate to one another. We then discuss each theme in greater detail.

Overview of Themes and Parameters
The analysis of the expert interviews revealed an intricate, interconnected network of design attributes for constructive assemblies, visualized in Figure 4. Attributes can be organized into a two-tier hierarchy: **themes** and **parameters**. We view **themes** (e.g. efficiency vs. exploration, and constrained vs. expressive) as higher-level attributes that primarily characterize the user experience, and can be *indirectly* influenced by the TUI designer through the *direct* manipulation of lower-level **parameters** (e.g. number of modules, module granularity). Themes cannot be considered in isolation from one another; rather, they are closely interrelated through a web of the parameters. By tuning parameters, designers can help shift their systems to better meet their design intentions. Four themes emerged from speaking with experts, which take on the form of bipolar axes:

- ***Efficiency vs. Exploration*** describes the extent to which users are encouraged to arrive at a specific configuration versus try out different physical arrangements of modules
- ***Repulsion vs. Attachment*** refers to the degree to which a user is inclined to either dispose of or keep the modules of a constructive assembly or assembled wholes
- ***Constrained vs. Expressive*** refers to the size of the design space covered by the constructive assembly
- ***Novice vs. Expert*** describes the extent to which the constructive assembly is geared towards novices or experts in the target problem domain.

The tight coupling of parameters and themes can be illustrated as follows. For instance, increasing both the number of possible connections and modular granularity (i.e. making modules smaller) would increase the complexity of the assembly. This would make it more difficult for novice users to handle, but would boost the degree of expressiveness of the constructive assembly. If this expressiveness is wielded skillfully by the user to create assemblies that are aesthetically pleasing, they may feel stronger attachment to their constructions. Consequently, these users would be more hesitant to disassemble them and would experiment less with different configurations.

Theme 1: Efficiency vs. Exploration
Efficiency and exploration represent two sides of the same coin. This theme (see Figure 5) captures this concept and describes to what degree a system encourages users in trying out different physical configurations versus arriving at a specific configuration of modules.

In general, making modules robust (E1, E8), cheap (E1) and easy to attach and detach from one another (E1) was said to encourage exploration. Furthermore, limiting the number of modules in a set can *"force people to modify it and destroy it"* (E1) and thereby drive exploration by necessity. Emphasis on physical over digital representations was said to encourage

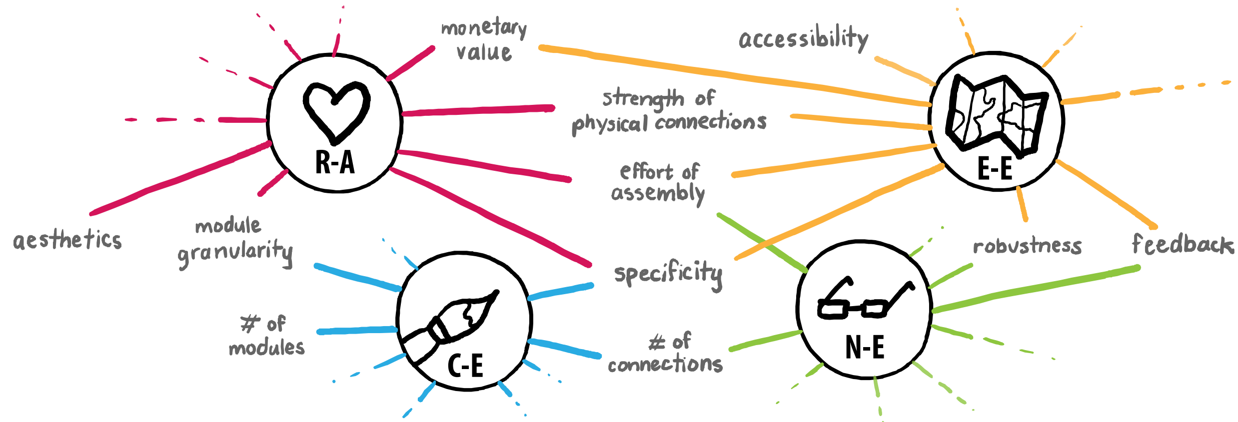

Figure 4. A figurative depiction of the network of the themes for constructive assemblies [Efficiency vs. Exploration (E-E), Repulsion vs. Attachment (R-A), Constrained vs. Expressive (C-E), and Novice vs. Expert (N-E)] and some of their associated parameters.

exploration since "*anything physical will be modified or will be used for something different*" (E1), while making physical modules more generic would have a similar effect and be a catalyst for *"creative misuse"* (E1). In contrast, working opposite to these suggestions would push a design towards the efficiency side of the axis.

Experts pointed out that a user's personality and mentality are also influential, albeit they lie outside the designers' control. *"Revolutionary thinking"* (i.e. thinking in a cycle of construction and deconstruction) and an attitude of *"not getting too precious about individual constructions"* motivates users to explore (E6), but there is ultimately "*variation within the users"* where some are *"super proactive and reconfigure a simple interface all the time"* while others are *"very lazy" (E8).*

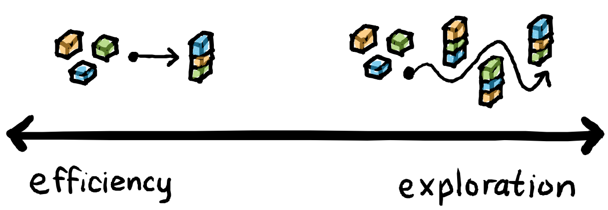

Figure 5. Efficiency vs. Exploration Axis.

Prevalence in Literature
The target problem to be addressed by the constructive assembly determines where the design should be situated along the axis. For instance, this theme has played an important role in past constructive assemblies such as with the caterpillar for storytelling in [3], or in Triangles [13] where designers intended for users to experiment with different configurations and topologies. In contrast, a constructive assembly designed to be used as a rig for 3D character animation [11] seeks to position itself on the efficiency side of this axis.

Theme 2: Repulsion vs. Attachment
This theme (see Figure 6) refers to the level of emotional attachment a constructive assembly fosters. Users can develop attachment to either the system as a material, or to constructed wholes. As mentioned by E6, *"people don't feel bad making a thing and breaking it apart. But they do feel really precious about the material itself,"* referring to how users have *"strong emotional reactions"* to supergluing LEGO® bricks such that blocks cannot be reclaimed (E6).

In general, it appears that user attachment to constructions largely hinges on whether they anticipate them to stay intact; weak physical connections and low material value and quality hampers feelings of attachment. E6 mentioned that people are *"much less precious"* about traditional building blocks "*because they don't hold themselves together,"* and see cheap Popsicle sticks as *"entirely fungible [and] completely disposable."* Conversely, having an aesthetic look and a more specific rather than general purpose nature can foster attachment. As stated by E2, *"if you have a beautifully designed object, you will form a very strong connection with it."* Stated by E6, *"people are going to have a more personal relationship with a specific teddy bear, as opposed to blocks that could be anything."* Additionally, the mere act of building was identified as a process that can impart significance to an assembled whole (E2, E7).

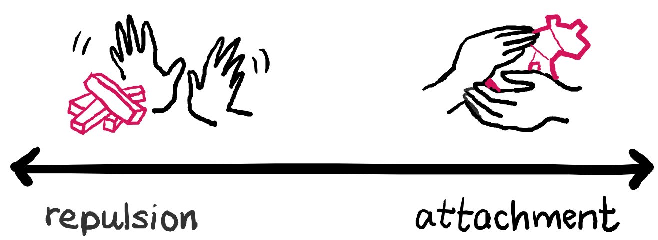

Figure 6. Repulsion vs. Attachment Axis.

Beyond designers' control, we can recognize that some people simply like to keep their constructions (E5). Time can also play a factor wherein attachment can develop retrospectively: *"a lot of the love for LEGO is retrospective … you remember afterwards all this time you spent doing things…"* (E6).

Prevalence in Literature
We observe that existing constructive assemblies do not span broadly across this design axis. With regards to attachment to constructive assemblies as a material, designers have predominantly strived to foster higher levels of emotional attachment, for instance, by leveraging *aesthetics* and incorporating a variety of pleasant colors [11,15,27].

The observed gaps along this axis can be exploited as new areas for research. In fact, we suspect that as the fabrication of modular and interactive pieces becomes cheaper and easier, more examples of one-time or 'disposable' constructive assemblies will emerge. For instance, we can consider a constructive assembly kit to help people physically express their ideas, which fosters low levels of attachment such that people dispose of it after use – like scrap paper. Conversely, with regards to fostering high levels of attachment for assembled wholes, one can envision assemblies for creating personalized robots as companions. In this case, the underlying design intention may be for people to keep assembled creations largely intact rather than completely disassemble them.

Theme 3: Constrained vs. Expressive

This theme (see Figure 7) refers to the size of the space of possibilities the constructive assembly can accommodate, and touches upon the degrees of freedom a system has.

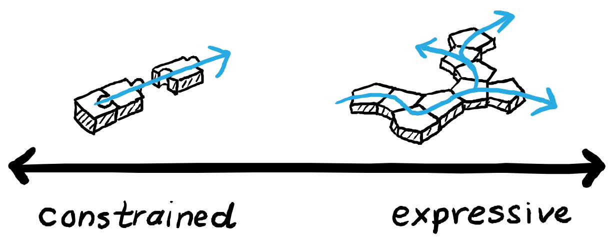

Figure 7. Constrained vs. Expressive Axis.

On the whole, a design can be made more expressive by introducing more physical and digital degrees of freedom. Specific examples include increasing the number of types of modules or the number of possible physical connections. Whilst not a constructive assembly, the reacTable [19] (an interactive surface, tabletop TUI for live music performance), was mentioned as allowing users to "*express their musical intention*" due the fact that it allows for "*many combinations … and so many configurations, which is virtually endless*" (E4). Increasing granularity (i.e. reducing module size) can also extend the expressive capacity of the system by offering users a finer degree of control and more means of physical manipulation: "*Clay is expressive and dynamic. You can hammer with a fist on it and make something with it, whereas you wouldn't do that with LEGO bricks*" (E4). Additionally, "hacking" can be seen as a user attempt to expand the expressiveness, for which designers can explicitly choose to support (E1), and the growing access to prototyping tools can help to fuel this (E4).

On a superficial level, it may seem that this theme mirrors the efficiency vs. exploration theme. However, we highlight the distinction that this theme concerns the design space covered by the constructive assembly system, whereas the other focuses on the degree to which the system encourages users to explore possible configurations within the design space afforded by the system. Whether designers aim for a more constrained or expressive system depends on the target problem. Expressivity can work to cater to different tastes (E3) and working needs (E1), whereas more constrained systems can aid users in arriving at desirable topologies sooner.

Prevalence in Literature

Many works from literature lean towards high expressivity, as in Triangles [13] or the emergent interfaces assembly in [7]. As the ability to quickly connect and disconnect pieces is generally conducive for experimentation and creativity, many designers wish to exploit this by designing their system to have a broad space within which users can explore different topologies [27]. However, designers have also worked to constrain their systems – oftentimes to teach topics that involve strict rules or syntax. Constraints help to enforce users to recognize and internalize these strict rules; constructive assemblies for teaching programming [15,25] or for producing code [6] for instance can be situated on the constrained side of the axis.

Theme 4: Novice vs. Expert

This theme (see Figure 8) refers to the extent to which the constructive assembly is designed to cater to novice or expert users in a target domain. Experts generally felt that existing work has a tendency to target novices and "*address a very early stage of [skill] development,*" (E4). Short projects, small teams, (E2) and a desire to limit the scope of work (E4) were said to be reasons for this.

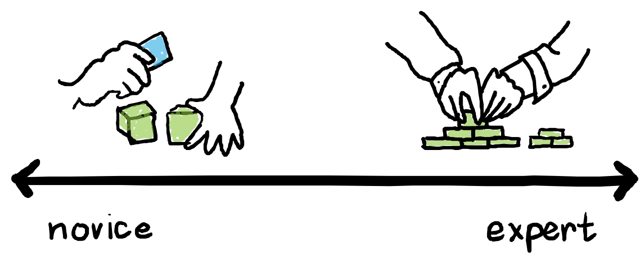

Figure 8. Novice vs. Expert Axis.

In general, experts expressed differing opinions on handling the novice-expert user spectrum. E5, E6, and E7 commented that tools should not be designed to handle both novices and experts. Conversely, comments from E1, E2, E3 and E5 suggested that it would be ideal for a TUI to cater and adapt to both. Hiding advanced functionality, allowing advanced functionality to be added (E1), and allowing learning aids to be removed (E1, E5) were suggested as ways to bridge the gap between different levels, in which case care should be taken to ensure that functionality is not then overlooked by users (E3).

Furthermore, novices and experts were said to have different sets of needs and desires. Novices value robustness and functional transparency, such that *"…no unexpected things can happen" (E1)*. E4 and E5 underlined the importance of feedback, as it "*helps [users] make sense of the response"* (E5). In some cases, physical rather than digital interactions were said to be helpful for novices, as many people are comfortable with modifying physical materials (E1). However, E2 and E4 cautioned against making constructive assemblies too simple. *"Easy is one thing people strive for, but for very wrong reasons. […] maybe it should be harder, not easier than using another, because most of the rewarding experiences we have are not easy ones"* (E2). Experts, on the other hand, were said to appreciate the ability to build constructions which do specific things (E1, E4), increase performance, and speed-up common actions (E1). Systems should also have enough depth such that experts can continue to *"discover new combinations [and] learn new things"* (E4).

Prevalence in Literature

This is a prominent theme and topic for discussion in the design of constructive assemblies. We note that in practice, no TUI targets a discrete level of expertise; rather, they always cater to a range of skill levels. However, in general, the bulk of systems focus on catering to novice users e.g. [6,35]. We suspect that the reason is two-fold. First, physical interfaces afford physical gestures or manipulations that many people are capable of performing and are used to doing in their everyday lives. Secondly, we suspect that this may be because many TUIs rely on a 1:1 representation of pieces and concepts, and thus rapidly become increasingly cumbersome to use as higher and more expert levels of complexity are approached. On the high end of the spectrum, we observe that few works have focused predominantly on catering to experts. This may begin to change however, as one work (a constructive assembly for 3D character animation [11]) has shown a significant breakthrough in tackling the practicality-complexity trade-off issue. It demonstrated how software can be effectively used to make constructive assemblies useful and appealing to expert users.

UTILITY OF THE FRAMEWORK: CASE STUDIES

In this section, we motivate the utility of the lifecycle model and the network of themes and parameters as a tool for critical reflection of existing constructive assemblies and as an aid to find directions for future iterations in their design. We examine two examples from literature through the lens of our framework, and also explore the concepts in the context of our own design case for a constructive assembly. In selecting these examples, we adhere to the definition of constructive assemblies in [7,17,28] as TUIs that have traditionally been inspired from LEGO and involve the interconnection of modular interactive elements that are manually or automatically put together to create more complex structures. Furthermore, these cases were chosen to include classical and modern examples that cover a diverse range of design goals and intentions, in order to demonstrate the broad applicability of the framework.

Case 1: Topobo

Topobo [27], is designed to teach children physics concepts, by enabling them to experiment in making biomorphic forms which can record and replay motion (see Figure 9, left). Its ***lifecycle*** comprises *construction (*piecing together modules), *usage* (observing assembled creations), *tweaking* (adding, removing, or programming pieces), and *destruction* (disassembling creatures). In evaluation, users who did incremental tweaking as opposed to constructed entire creations before programming had greater success in making creatures that were balanced and could walk. As visualized in Figure 10, left, the system supports high degrees of ***exploration***; generally, the *immediacy* of the interface in responding to user-manipulation encourages rapid experimentation and tweaking. However, a software bug of one module had detracted from

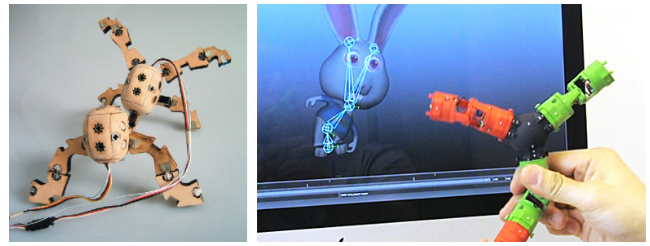

Figure 9. Examples of constructive assemblies: an early version of Topobo [28] (left), and a tangible and modular input device for the animation of 3D characters [10] (right).

this as it sometimes caused erratic behavior and gave people the fear of broken parts. The system appeals to a fairly high degree of ***attachment***, as it *reflects qualities of the user;* creatures tend to exhibit "emotionally engaging motions" as they mirror users' body movements. It is also highly ***expressive*** as it supports a wide breadth of possible topologies given a *high number of modules* (13) which were furthermore *programmable* to have unique motion patterns. The system primarily supports ***novices*** as pieces are generally *robust,* have *guides and scaffolding* integrated into them, and have a *simple* one-button interface for programming. However, the depth of expression afforded by the system expands its reach beyond novice users to also reach slightly more expert users.

Case 2: Tangible and Modular Animation Rig

This system uses a constructive assembly as a physical input device and rig for posing and animating 3D characters [11] (see Figure 9, right). The ***lifecycle*** for this assembly is straightforward, and comprises *creation* (assembling of an entire rig according to a software-generated suggestion), *usage* (repositioning the rig into different poses), optionally *storage* (keeping the rig intact for subsequent sessions), and *destruction* (disassembling the rig into constituent pieces for use in new rigs.) As seen in Figure 10, middle, this work focuses on supporting ***efficiency***. Software is used to bypass user experimentation, such that users can arrive at an effective configuration of modules to model a 3D character. The system features modules with pleasing *aesthetics,* fostering a moderate degree of ***attachment*** not for particular configurations but for the system as a tool. The system also features a high degree of ***expressiveness***, which is necessary for it to accommodate a large design space of possible 3D characters. This is achieved by designing modules to support a high *number of connections*, which are also *physically dynamic* and afford *movement.* It caters well to both ***novice and expert*** users. Direct manipulation makes interaction fairly *simple* which is good for novices, while it also offers experts a significant *speed-up* through the use of computational optimization and modelling.

Case 3: Designing an Emotion Music Player

We implemented a constructive assembly (see Figure 11) to allow users to find and play music that complements their changing moods. Different vertical configurations of colored LEGO bricks atop a monitor, detected with a depth-sensing camera, determine what music is played. To use the system, users first map colors to moods using an onscreen editor (e.g.

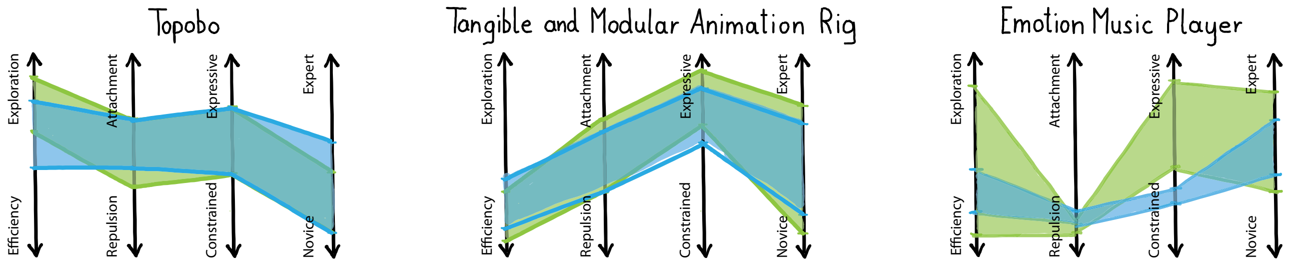

Figure 10. Mapping examples of constructive assemblies along the four design axes. Blue marks the current system's qualities; green indicates a direction for future iterations of the respective systems.

yellow to 'happy', green to 'relaxed'). Thereafter, adding or removing blocks to a stack increases or decreases a specific mood's weighting respectively. The system has a ***lifecycle*** comprising *creation* (assembly of stacks), *usage* (listening to the triggered music), *tweaking* (adding or removing blocks), *storage* (keeping a configuration constant while away from the desk) and *destruction* (disassembly of the stacks). The system should span both ***efficiency and exploration***, where users can find new music tracks that correspond with their mood. However, the limited number of colors hinders the ease with which a user can navigate the broad search space of music. This mismatch can be detected along the first axis of Figure 10, right. Offering *more physical means* to interact with music search parameters may improve this. The *generic* appearance discourages ***attachment*** to particular configurations and instead encourages users to experiment with different ones. As the current implementation only offers two *degrees of freedom* (colors, number of blocks), it appears to have an insufficient degree of ***expressiveness*** to cover the music space (see Figure 10, right). Introducing more (e.g. width or color sequences) could help to close the gap between the prototype and the design intentions. While the system is intended to appeal to a broad spectrum of users, it could be improved to better cater to both novices (as it has low *functional transparency* or unclear mapping between emotions and colors) and experts (as it lacks a means to formulate *specific* search queries).

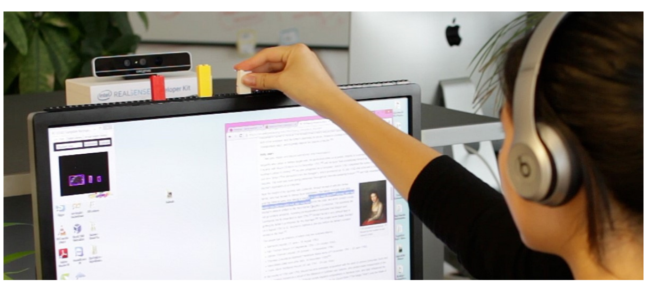

Figure 11. Emotion Music Player: users assemble stacks of blocks to control music to match their changing moods.

Case Studies Discussion

We applied our framework to different cases based on examples from literature and our own design study. The diversity of the examples covered highlight the framework's broad applicability, and the case studies illustrate how the framework can be used to critically analyze works that have very different design intentions, in order to provide directions for improvement that align appropriately with these intentions.

Applying the lifecycle model can help break down user-interactions to study them in greater depth. For instance, we can observe that Topobo's strength lies in supporting rapid *tweaking*. In contrast, the *storage phase* of the animation rig could be enhanced with the ability to reinstate past poses, since users are likely unable to complete animation tasks in a single session. The design axes helped to frame design choices and isolate both where designs succeed and fall short of their intentions (as illustrated in Figure 10). They also help to pinpoint parameters which may help push the design of a system closer towards its intended purpose.

CONCLUSION AND FUTURE WORK

While many established frameworks enhance our overall understanding of TUIs, few works have focused on highlighting key considerations for taking a constructive assembly approach to TUI design. In this paper, based on an analysis of data elicited through extensive interviews with world-class academic and industry TUI experts, we presented a framework for constructive assemblies. We presented benefits and drawbacks, established a lifecycle model, and unveiled key design themes pertaining to constructive assemblies alongside their associated design parameters. We applied our framework to our own work and examples from literature, and demonstrated its utility for supporting critical reflection and discussion of design choices and their implications. While one can consider it a limitation that systems cannot be precisely quantified along the four themes, the process of mapping them along the axes helps to focus one's attention on adjusting qualities that have a direct impact on the user experience for the system. Given its utility, we would like to, in future, continue using it for critical reflection to drive the improvement of constructive assembly designs. Moreover, we see that there may be opportunities to expand this framework; as more instances of constructive assemblies arise, it could be refined to contain more attributes, which in turn could further strengthen its utility for designers and practitioners making such TUIs. As some of the presented concepts appear to have use beyond constructive assembly TUIs, it would also be interesting to explore its applicability in a broader context for future research.

ACKNOWLEDGEMENTS

We thank our experts for their time in interviews, the authors who allowed us to picture their work, and Kathrin Probst for her valuable feedback and idea to present charts. This work received the funding support of NCBiR, FWF, SNSF, ANR, and FNR in the framework of the ERA-NET CHIST-ERA II.

REFERENCES

1. R. Aish. 1979. 3D input for CAAD systems. Computer-Aided Design 11, 2: 66–70. http://doi.org/10.1016/0010-4485(79)90098-8
2. Robert Aish and Peter Noakes. 1984. Architecture without numbers — CAAD based on a 3D modelling system. Computer-Aided Design 16, 6: 321–328. http://doi.org/10.1016/0010-4485(84)90116-7
3. Mike Ananny. 2002. Supporting children's collaborative authoring: practicing written literacy while composing oral texts. 595–596. Retrieved September 23, 2015 from http://dl.acm.org/citation.cfm?id=1658616.1658739
4. David Anderson, Jonathan S. Yedidia, James L. Frankel, et al. 2000. Tangible Interaction + graphical interpretation: a new approach to 3D modeling. Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00, ACM Press, 393–402. http://doi.org/10.1145/344779.344960
5. Richard Borovoy. 1996. Dr. LEGOHead, Ph.D.: a real-world story construction environment. 552. Retrieved September 23, 2015 from http://dl.acm.org/citation.cfm?id=1161135.1161220
6. David Bouchard and Steve Daniels. 2015. Tiles that Talk: Tangible Templates for Networked Objects. Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14, ACM Press, 197–200. http://doi.org/10.1145/2677199.2680607
7. Alexandru Dancu, Morten Fjeld, Catherine Hedler, et al. 2015. Emergent Interfaces. Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '15, ACM Press, 451–460. http://doi.org/10.1145/2702613.2732509
8. Kenneth P. Fishkin. 2004. A taxonomy for and analysis of tangible interfaces. Personal and Ubiquitous Computing 8, 5: 347–358. http://doi.org/10.1007/s00779-004-0297-4
9. J.H. Frazer, J.M Frazer, and P.A Frazer. 1980. Intelligent physical three-dimensional modeling system. Computer Graphics Conference, vol. 80, 359–370.
10. John Frazer. 1995. An Evolutionary Architecture. Architectural Association, London.
11. Oliver Glauser, Wan-Chun Ma, Daniele Panozzo, Alec Jacobson, Otmar Hilliges, and Olga Sorkine-Hornung. 2016. Rig animation with a tangible and modular input device. ACM Transactions on Graphics 35, 4: 1–11. http://doi.org/10.1145/2897824.2925909
12. Jennifer W. Glos and Justine Cassell. 1997. Rosebud. CHI '97 extended abstracts on Human factors in computing systems looking to the future - CHI '97, ACM Press, 359. http://doi.org/10.1145/1120212.1120433
13. Matthew G. Gorbet, Maggie Orth, and Hiroshi Ishii. 1998. Triangles. Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '98, ACM Press, 49–56. http://doi.org/10.1145/274644.274652
14. Lars Erik Holmquist, Johan Redström, and Peter Ljungstrand. 1999. Token-Based Access to Digital Information. Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, Springer-Verlag, 234–245.
15. Michael S. Horn, Erin Treacy Solovey, and Robert J. K. Jacob. 2008. Tangible programming and informal science learning. Proceedings of the 7th international conference on Interaction design and children - IDC '08, ACM Press, 194. http://doi.org/10.1145/1463689.1463756
16. Eva Hornecker and Jacob Buur. 2006. Getting a grip on tangible interaction. Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06, ACM Press, 437. http://doi.org/10.1145/1124772.1124838
17. Hiroshi Ishii. 2008. Tangible bits: beyond pixels. Proceedings of the 2nd international conference on Tangible and embedded interaction - TEI '08, ACM Press, xv. http://doi.org/10.1145/1347390.1347392
18. Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, et al. 2008. Reality-based interaction. Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08, ACM Press, 201. http://doi.org/10.1145/1357054.1357089
19. Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable. Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07, ACM Press, 139. http://doi.org/10.1145/1226969.1226998
20. Yoshifumi Kitamura, Yuichi Itoh, Toshihiro Masaki, and Fumio Kishino. 2000. ActiveCube: a bi-directional user interface using cubes. KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies., IEEE, 99–102. http://doi.org/10.1109/KES.2000.885768
21. Rong-Hao Liang, Liwei Chan, Hung-Yu Tseng, et al. 2014. GaussBricks. Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14, ACM Press, 3153–3162. http://doi.org/10.1145/2556288.2557105

22. Paul Marshall. 2007. Do tangible interfaces enhance learning? Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07, ACM Press, 163. http://doi.org/10.1145/1226969.1227004
23. Fred Martin and Mitchel Resnick. 1993. LEGO/Logo and Electronic Bricks: Creating a Scienceland for Children. In Advanced Educational Technologies for Mathematics and Science, David L. Ferguson (ed.). Springer Berlin Heidelberg, 61–89. http://doi.org/10.1007/978-3-662-02938-1_2
24. Ali Mazalek and Elise van den Hoven. 2009. Framing tangible interaction frameworks. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 23, 3: 225. http://doi.org/10.1017/S0890060409000201
25. Timothy S. McNerney. 2004. From turtles to Tangible Programming Bricks: explorations in physical language design. Personal and Ubiquitous Computing 8, 5: 326–337. http://doi.org/10.1007/s00779-004-0295-6
26. Hyunjoo Oh and Mark D. Gross. 2015. Cube-in. Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14, ACM Press, 383–386. http://doi.org/10.1145/2677199.2680597
27. Hayes Solos Raffle, Amanda J. Parkes, and Hiroshi Ishii. 2004. Topobo. Proceedings of the 2004 conference on Human factors in computing systems - CHI '04, ACM Press, 647–654. http://doi.org/10.1145/985692.985774
28. Anne Roudaut, Rebecca Reed, Tianbo Hao, and Sriram Subramanian. 2014. Changibles. Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14, ACM Press, 2593–2596. http://doi.org/10.1145/2556288.2557006
29. Philipp Schoessler, Daniel Windham, Daniel Leithinger, Sean Follmer, and Hiroshi Ishii. 2015. Kinetic Blocks. Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15, ACM Press, 341–349. http://doi.org/10.1145/2807442.2807453
30. Eric Schweikardt and Mark D. Gross. 2006. roBlocks. Proceedings of the 8th international conference on Multimodal interfaces - ICMI '06, ACM Press, 72. http://doi.org/10.1145/1180995.1181010
31. Orit Shaer. 2009. Tangible User Interfaces: Past, Present, and Future Directions. *Foundations and Trends in Human–Computer Interaction* 3, 1–2: 1–137. http://doi.org/10.1561/1100000026
32. Orit Shaer, Nancy Leland, Eduardo H. Calvillo-Gamez, and Robert J.K. Jacob. 2004. The TAC paradigm: specifying tangible user interfaces. *Personal and Ubiquitous Computing* 8, 5: 359–369. http://doi.org/10.1007/s00779-004-0298-3
33. Hideyuki Suzuki and Hiroshi Kato. 1995. Interaction-level support for collaborative learning. *The first international conference on Computer support for collaborative learning - CSCL '95*, Association for Computational Linguistics, 349–355. http://doi.org/10.3115/222020.222828
34. B. Ullmer and H. Ishii. 2000. Emerging frameworks for tangible user interfaces. *IBM Systems Journal* 39, 3.4: 915–931. http://doi.org/10.1147/sj.393.0915
35. Peta Wyeth and Helen C. Purchase. 2002. Tangible programming elements for young children. *CHI '02 extended abstracts on Human factors in computing systems - CHI '02*, ACM Press, 774. http://doi.org/10.1145/506443.506591
36. Jamie Zigelbaum, Michael S. Horn, Orit Shaer, and Robert J. K. Jacob. 2007. The tangible video editor. *Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07*, ACM Press, 43. http://doi.org/10.1145/1226969.1226978
37. Oren Zuckerman, Saeed Arida, and Mitchel Resnick. 2005. Extending tangible interfaces for education. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, ACM Press, 859. http://doi.org/10.1145/1054972.1055093
38. littleBits Electronics. Retrieved July 31, 2015 from http://littlebits.cc/