Molecular HCI: Structuring the cross-disciplinary space of modular shape-changing user interfaces

Laura Pruszko, Céline Coutrix, Yann Laurillau, Benoît Piranda, Julien Bourgeois

▶ **To cite this version:**

Laura Pruszko, Céline Coutrix, Yann Laurillau, Benoît Piranda, Julien Bourgeois. Molecular HCI: Structuring the cross-disciplinary space of modular shape-changing user interfaces. Proceedings of the ACM on Human-Computer Interaction , In press. hal-03215058v1

HAL Id: hal-03215058

https://hal.science/hal-03215058v1

Submitted on 3 May 2021 (v1), last revised 12 May 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL
archives-ouvertes.fr

Molecular HCI: Structuring the cross-disciplinary space of modular shape-changing user interfaces

Laura Pruszko, Céline Coutrix, Yann Laurillau, Benoît Piranda, Julien Bourgeois

▶ **To cite this version:**

Laura Pruszko, Céline Coutrix, Yann Laurillau, Benoît Piranda, Julien Bourgeois. Molecular HCI: Structuring the cross-disciplinary space of modular shape-changing user interfaces. Proceedings of the ACM on Human-Computer Interaction , Association for Computing Machinery (ACM), In press. hal-03215058

HAL Id: hal-03215058
https://hal.archives-ouvertes.fr/hal-03215058

Submitted on 3 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Molecular HCI: Structuring the cross-disciplinary space of modular shape-changing user interfaces

LAURA PRUSZKO, Université Grenoble Alpes, France
CÉLINE COUTRIX, CNRS & Université Grenoble Alpes, France
YANN LAURILLAU, Université Grenoble Alpes, France
BENOIT PIRANDA, Univ. Bourgogne Franche-Comté, Institut FEMTO-ST, CNRS, FRANCE
JULIEN BOURGEOIS, Univ. Bourgogne Franche-Comté, Institut FEMTO-ST, CNRS, FRANCE

Shape-changing User Interfaces attract growing interest in Human-Computer Interaction. Modular robotics offer a great opportunity for their implementation. However, the current theoretical and technical advances of modular robotics are fragmented and little centered on the user. To unify existing work and center future research on the user, we perform a systematic literature review enabling us to build a unifying space for the design of modular shape-changing user interfaces. Our aim is to bridge the gap between HCI and robotics. Towards this aim, we conduct a thorough cross-disciplinary survey to propose: 1) a set of design properties at the scale of the interface (macro-scale) and at the scale of the modules (micro-scale) and 2) the impact of these properties on each other. We relate properties of different domains and identify inconsistencies to structure the design space. This paper can be used to describe and compare existing modular shape-changing UIs and generate new design ideas by building upon knowledge from robotics and HCI.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; **HCI theory, concepts and models**; **Human computer interaction (HCI)**.

Additional Key Words and Phrases: Shape-changing interfaces, Modular user interfaces, Properties, User Interface properties, Modules properties, Conceptual work.

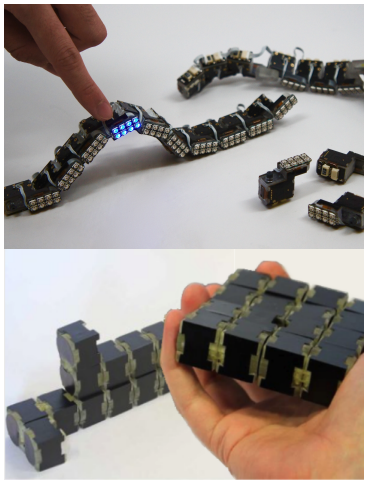
ACM Reference Format:
Laura Pruszko, Céline Coutrix, Yann Laurillau, Benoit Piranda, and Julien Bourgeois. 2021. Molecular HCI: Structuring the cross-disciplinary space of modular shape-changing user interfaces. 1, 1 (April 2021), 31 pages. https://doi.org/xx.xxxx/xxxxxxx

1 INTRODUCTION

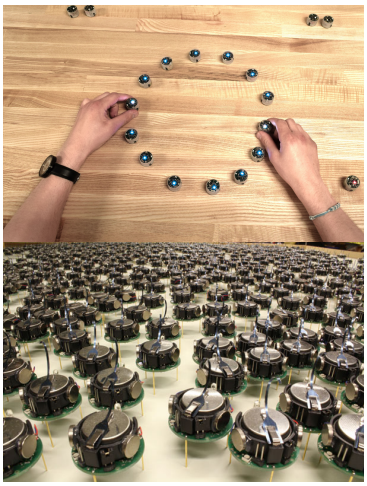
Shape-changing User Interfaces (UIs) are tangible interfaces able to change their physical shape to support input, output or both. They leverage the benefits of physicality from tangible UIs and the benefits of flexibility from graphical UIs. For this reason, they attract growing interest in Human-Computer Interaction (HCI) since 2004 [65]. Shape-changing UIs enable, e.g., the unique support of adaptative affordances, the augmentation of users or the communication of information [2].

Authors' addresses: Laura Pruszko, Université Grenoble Alpes, France, laura.pruszko@univ-grenoble-alpes.fr; Céline Coutrix, CNRS & Université Grenoble Alpes, Grenoble, France, celine.coutrix@univ-grenoble-alpes.fr; Yann Laurillau, Université Grenoble Alpes, France, yann.laurillau@univ-grenoble-alpes.fr; Benoit Piranda, Univ. Bourgogne Franche-Comté, Institut FEMTO-ST, CNRS, Montbéliard, FRANCE; Julien Bourgeois, Univ. Bourgogne Franche-Comté, Institut FEMTO-ST, CNRS, Montbéliard, FRANCE.

(a) Example lattice implementations.
Top: ATRON [76].
Bottom: M-Blocks [89].



(b) Example chain implementations.
Top: chainFORM [72].
Bottom: Cubimorph [91].



(c) Example swarm implementations.
Top: Zooids [57].
Bottom: Kilobot [93].

Fig. 1. Examples of modular shape-changing UIs and the three main types of architecture: (a) lattice, (b) chain and (c) swarm.

An approach to implement shape-changing UIs is based on modular robots. Examples include Zooids [57] (Figure 1c, top) which consist of cylindrical wheeled modules enabling 2D reconfiguration on flat surfaces. Zooids applications include, e.g., reconfigurable physical scatterplots. Another example is chainFORM [72] (Figure 1b, top) which consists of chained rectangular modules supporting 3D reconfiguration. ChainFORM applications include, e.g., reconfigurable wearable haptic displays. Such *modular robot* is defined as a large number of small scale robotic modules that can spatially rearrange (by themselves or not). A *robotic module* is defined as a microelectromechanical system embedding computational capabilities. In this paper, we define *modular shape-changing UIs* as shape-changing UIs made of a large number of such robotic modules. Such modular shape-changing UIs are able to compute collectively to support interaction, e.g., to provide a visual or haptic display through the reconfiguration of their shape or to sense the user's touch location.

Modular robotics offer great perspectives to address current challenges such as scalability [2], sustainability [2], robustness [122], cost [122] and versatility [122]. Moreover, modular robotics offer a great opportunity for *modularity*- and *porosity*-changing UIs, in addition to other changes in shape. Modularity is the ability of an object to be split in at least two parts and (re)combined while maintaining its original functionality. Porosity is the ratio of the area of perforated parts to the total area of the shape. Modularity and porosity are key features in shape-changing UIs taxonomies (*Modularity* [47] or *Adding/Substracting* [87], and *Porosity* [47] or *Permeability* [87]). Modularity and porosity are difficult to implement with other approaches. For instance, the pneumatically actuated air pouches of PneUI [120] can dynamically open/close a lid on top of a hole to support limited changes in porosity. Such change of shape is pre-programmed at design stage and cannot be modified during interaction. In contrast, modular shape-changing UIs can provide a larger range of porosity, even if there were not planned at design stage.

Despite their promises, the HCI community seldom leverage modular robotics for shape-changing UIs. The problem is that we lack knowledge in 1) how to build modular robotic systems and 2) how to build upon existing knowledge in robotics that is not user-centered.

First, modular shape-changing UIs are difficult to implement for HCI researchers. Prototyping shape-changing UIs in general requires, among others, complex skills to leverage current knowledge in materials, electronics and mechanics, whereas the HCI community is typically skilled in software programming or simple electronics [2]. We focus on hardware properties in this paper since hardware prototyping of *modular* shape-changing UIs is the current greatest challenge: the hardware is still a research topic, and needs to become smaller, lighter, stronger and faster in order to be comparable with the capabilities of other approaches [2]. They are therefore seldom studied in HCI, with few configurations of robots and scenarii explored (e.g., [28, 57, 73, 91]). The existing systems are built in an ad-hoc manner, and few design choices are user-centered and/or documented. When researchers need to build a new system, they might have to start from scratch, and deal with the same design issues again. In these conditions, it is hard to explore design properties in a systematic way, and one can easily find a better solution after the implementation is finished. Existing HCI tools for the design of shape-changing UIs (e.g., surveys, taxonomies, or challenges) globally consider all shape-changing UIs, without taking into account the specificity of modular robotics. While there are other benefits in technology-agnostic design, it is however hard to know which particular robotic modules allow the implementation of a design. Existing HCI design rationales for modular shape-changing UIs each addresses a local set of design properties, rather than providing a global viewpoint on modular interfaces (e.g., handheld chain interfaces [91] or tabletop swarm interfaces [57]).

Second, while modular robotics show great promises for HCI, the current theoretical and technical advances of modular robotics are little centered on the user. Robotics researchers have conducted extensive work on modular robots since 1990 [23] and proposed several advanced robot configurations. However, their implementations are mostly designed for construction or locomotion rather than user interaction [91]. Thus, robotics tools for the design of modular robots are not user-centered, nor take into account the impact of the technical aspects of the modules on the interaction with the user. As a consequence, there is no coherent set of properties in the literature taking into account the specificities of modular shape-changing UIs and their impact on the interaction. However, specifying user-centered properties early on to inform the research and the design of an interactive system is a key element of user-centered design [39]. As a consequence, specifying user-centered properties is highly important to further enable HCI research on modular robots to support modular shape-changing UIs.

Providing a tool for the systematic exploration of the design properties of modular shape-changing UIs is challenging as the knowledge bridges both HCI and robotics communities, and covers decades of research.

In this paper, we bridge the gap between the need in HCI for modular shape-changing UIs and the experience in robotics in building modular systems. We build upon the literature in both domains to propose:
(1) A set of **properties at the scale of the interface (*macro scale*)** (Figure 2, left),
(2) A set of **properties at the scale of the modules (*micro scale*)** (Figure 2, right), and
(3) An analysis of their dependencies.

Our contribution lies in the identification, selection and structuring of a unified set of properties. These properties bridge the gap between the HCI and robotics fields. Our work provides a tool allowing the *description*, the *evaluation* and the *generation* (i.e. the help for novel design) [6] of modular shape-changing UIs. The HCI community can readily use our work to inform design choices and consider alternatives. The community can build upon this work to research usable modular shape-changing UIs.

2 BACKGROUND

The robotics field has researched modular systems since the 1990s, whereas HCI has explored them, less extensively, since 2007 (Siftables [64] was modular without actuation). As a consequence, robotics have already proposed

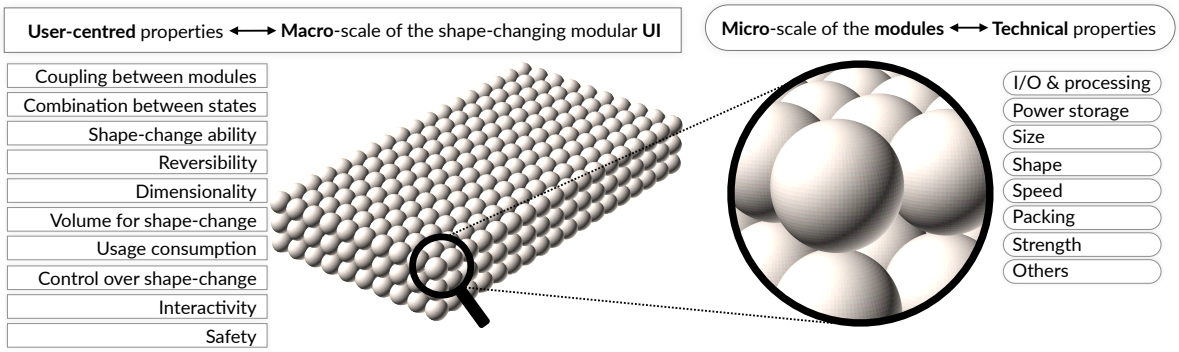


Fig. 2. Our structured space of properties: (left & green) user centered properties at the macro scale of the shape-changing modular UI, and (right & yellow) technical properties at the micro-scale of the modules.

many systems. We explain here their different types of *architectures* and *reconfigurations* that we will link in the paper to our user-centered properties.

2.1 Architectures

Architecture include the relative physical geometric arrangement of the modules [108] (e.g., lattice, chain, swarm/mobile and hybrid) and the homogeneity of the modules.

Lattice modules are arranged on a regular 2D- or 3D-grid structure called a lattice. For example, 3D M-Blocks [90] (Figure 1a, bottom) are cubic robotic modules that can roll on each other with permanent magnets and jump with flywheels (as shown in Figure 1a, bottom) to change the overall shape. Another example are the spherical modules in ATRON (Figure 1a, top) that can also rearrange themselves, but in their case through latching to their neighbours with mechanical clamps. Their goal is shape reconfiguration, locomotion (e.g., snake-like robot, wheeled-robot or legged robot) and manipulation (e.g., robot arm). We find different lattice geometries, e.g., face-centered cubic lattice [82] (Figure 5b), simple cubic lattice (Figure 5a) or hexagonal lattice [122]. The displacement of modules during reconfiguration is dependent on this geometry. Compared to other architectures, both the control and motion of each module can be executed in parallel, allowing for faster and easier reconfiguration. Lattice systems could be used in the future to enable, e.g., physical computer-aided design, as they allow as many shapes as play-doh. However, it is currently difficult to conduct HCI research with lattice systems: most contributions are simulations (e.g., [111]) or concepts (e.g., [83]), and working prototypes are few and/or too early (e.g., [25, 76, 90]). Few working prototypes supports user input, but their modules cannot move by themselves [53, 62].

Chain modules are connected together following a string (e.g., ChainFORM [72] or Cubimorph [91] in Figure 1b) or a tree topology [122]. The displacement of modules is serial. Unlike lattice implementations, chained modules do not have to fully stick to the face of a neighbor , but they can be stable in any position between minimum and maximum rotation of the motor joining the two modules. Compared to other architectures, the reconfiguration is more difficult to control, represent and analyse [110, 122]. Chain-based systems are already used for HCI research (e.g., [72, 73]).

Swarm modules can move independently. Examples include Zooids [57] shown in Figure 1c (top). Another type of modules, called **"mobile"** in robotics, can in addition latch and delegate their mobility to their neighbor (e.g., [23]). Examples of such additional capability include the second version of Zooids [126] where modules can stack and be moved by their supporting neighbor. Swarm UIs are already used for HCI research (e.g., [10, 57]).

Hybrid systems mix the previously mentioned architectures. For example, we find in the literature several instances of chain×lattice [43, 98, 121] and, less common, chain×mobile [35] or chain×lattice×mobile [16].

Complementarily, modular shape-changing UIs are either **homogeneous**, i.e. all modules have the same design (hardware *and* software) or **heterogeneous**, i.e. modules have different designs (hardware *and/or* software). Typically, heterogeneous systems are composed of sub-groups of homogeneous modules [19]. Most systems are homogeneous, to ease mass production, self-repair and self-reconfiguration [70, 71]. However, heterogeneous systems offer interesting perspectives as individual modules in a same system can embed different sensing and computational capabilities [85].

2.2 Reconfigurations

Reconfigurations show different abilities (e.g., self-reconfiguration or self-(dis)assembly) and different approaches (stochastic or deterministic).

While self-(dis)assembly is the *(dis)connection* of modules (i.e. actuated *Modularity* [47]), self-reconfiguration is the movement of *already assembled* modules.

Self-reconfiguration allows for autonomous shape-change [91, 95]. Parts or all of the modules composing the interface move to change from an initial shape to a target shape. This is the most common type of reconfiguration in the literature and across architectures, ranging form lattice (e.g., [9, 90]), chain (e.g., [91, 100]), swarm (e.g., [57, 93]), mobiles (e.g., [23]), and hybrid implementations (e.g., [43, 121]).

Self-assembly modules are initially detached from each another (e.g., no *initial shape* but a set of unlatched robots in an unknown configuration). They individually move and latch to assemble into a larger target shape, which has greater capabilities than the individual modules [35]. Self-assembly systems are either hybrid (e.g., swarm×chain [35], mobile×chain [16], pin×lattice [106]) or use stochastic reconfiguration (as presented below) (e.g., [4, 25, 32]).

Self-disassembly starts from initially assembled modules, and unlatch –i.e. let go of– parts of the structure to achieve a more interesting and functional one [25, 26]. Self-disassembly is implemented through lattice-based architectures (e.g., [25, 26]).

Future types of reconfiguration are studied mostly as a vision since they present several technical challenges. For example, self-repairing systems could recover from damages by replacing faulty units [1]. Self-replicating systems could take one step further, being able to build copies of themselves [122, 127].

Complementarily, we found two approaches for modular reconfiguration: deterministic vs. stochastic.

Deterministic reconfiguration relies on the ability of the system to know or compute the location of all modules, in order to achieve a target shape. Deterministic reconfiguration is predominant across the literature: 153 implementations among the 159 we studied use deterministic reconfiguration.

Stochastic reconfiguration relies on the environment to move the robots, e.g., a moving support surface (e.g., [4, 25, 32]), in order to achieve a target shape. The reconfiguration relies on statistical processes [122]: in the case of self-assembly, when two modules come in contact, they share their internal state to evaluate whether they are intended to be neighbours to achieve a given overall target structure. If they do not, they repel each other [32]. The structure grows gradually, "in an organic manner" [25] until completion of the target structure.

Deterministic and stochastic reconfigurations are not mutually exclusive. For example, a system can use stochastic self-assembly to build an initial block of modules and then use deterministic self-disassembly to detach unwanted modules and reach a more complex final shape [25].

3 RESEARCH METHODOLOGY

We conduct a systematic review of the literature from both HCI and robotics. We follow a four steps methodology as in previous work [59, 116]: identification, screening, eligibility and inclusion.

3.1 Identification

This step follows the rules described in [116]. We used the advanced search feature, in four major computer science digital libraries: ACM Digital Library, IEEEXplore, Springer Link, and Science Direct. To ensure we did not miss references from other publishers, we additionally search for references on Google Scholar. We performed the search on all available data (e.g., title, keywords, abstract, full text).

Multi-disciplinarity renders choosing relevant and comprehensive keywords difficult. First, HCI and robotics do not share the same vocabulary (e.g., *"shape-changing UI"* in HCI and *"programmable matter"* in robotics). Second, there is a wide range of keywords, from the general modular aspect (e.g., *"modular interface"*) to specific technical aspects (e.g., *"self-assembly"*) of the interface. To match this great diversity in keywords and make sure we do not miss any relevant paper, we chose to run two queries: (1) one using *general keywords* describing the modular aspect of the system and (2) one using *technical keywords* describing the architecture and type of reconfiguration.

General query. We combined the following general keywords about modular interfaces in both HCI and robotics domains: *modular AND ("programmable matter" OR "modular robot" OR "modular interface" OR "shape-changing interface")*. This ensures that each general keyword includes the modular aspect. We obtained 277 results on ACM Digital Library, 848 results on IEEEXplore, 1,672 results on Springer Link, 1,036 results Science Direct, and 11,600 results on Google Scholar.

Technical query. Our second query is as follow: *robot AND ("self-reconfigurable" OR "self-assembly" OR "self-disassembly" OR swarm OR chain OR lattice)*. We discarded the keyword *"hybrid"* as papers describing hybrid systems further specify the architecture types (e.g., lattice×chain, chain×swarm). We obtained 4,640 results on ACM Digital Library, 7,232 results on IEEEXplore, 44,361 results on Springer Link, 50,93 results Science Direct, and 730,000 results on Google Scholar.

For each query and on each library, we displayed the results by decreasing order of relevance. We included all references if the query returned less than 400. If the query returned more than 400 references, we included the first 400 and performed a manual check of the next 200 to ensure we did not miss relevant papers. Doing so, we obtained 1,725 references from the general query and 1,836 references from the technical query, i.e. a total of 3,561 references.

3.2 Screening

As in [24], we screened the title and abstract of each paper to remove duplicates and determine their relevance to the research question. However, we could not find any irrelevant papers at this phase. Indeed, assessing relevance solely through title and abstract proved challenging. For example, the term "modular robot" is largely used for industrial robot manipulators which are outside the scope of this paper. We could not leave out these papers at the screening phase. After removing 273 duplicates, we obtained a total of 3,288 references.

3.3 Eligibility

Following [116], we evaluated each paper on their *form* and *content*. Concerning *form*, we only retained papers written in English and published in a peer reviewed venue. We did not consider PhD and Master theses, as we expect such research to be identified in our identification step through their resulting peer-reviewed publications.

Concerning *content*, we only retained papers matching the two following selection criteria:

(1) **Scope of contribution**: We considered papers that study modular shape-changing UIs, that we defined in the introduction as UIs made of a large number of robotic modules, i.e. microelectromechanical system embedding computational capabilities, allowing them to change shape (by themselves or not).

(2) **Type of contribution**: We considered papers presenting an implementation, a tool for its design, or a survey of implementations. Among papers presenting an implementation, we considered papers either presenting

a working or conceptual implementation. Among papers presenting an implementation, we only considered papers presenting hardware or interaction design contributions. Work on reconfiguration software and algorithms were left apart for this first version of our space of properties, and kept for future work.

We discarded 2,708 ineligible papers, leaving 580 references to be included.

3.4 Inclusion

Some references that we know to be relevant did not show up with our queries: two implementations [23, 91], a paper on human-swarm interaction using the Zooids plateform [51] and a taxonomy [108]. First, to ensure that we do not miss any other relevant implementations, we cross-checked the implementations we found with the ones presented in the surveys and related work from our corpus. We found seven additional implementations, which we further included [23, 37, 54, 69, 91, 97, 115]. Second, we ran the query *"Human-Swarm Interaction"* through the same databases we used for the rest of our corpus. After removing duplicates and ineligible papers, we found 17 additional references which we further included. Finally, we added the taxonomy paper [108]. We hypothesize this paper did not appear in the results of our queries because it is recent (2020).

With the inclusion of these 25 papers, we obtained a corpus of 605 papers. However, a single implementation, e.g., different versions, can be described in several papers. For this reason, we found 159 unique hardware or conceptual implementations presented in a total 485 papers.

We also found 92 papers describing tools for design (e.g., taxonomies) and 39 surveys of existing implementations. A single paper can present several of these types of contribution.

3.5 Corpus analysis

Our aim is to unify and structure the existing design properties and re-center them around the user. We took a 6-steps approach to analyze our corpus. We (1) identified the relevant properties guiding the design of existing implementations or proposed in the tools for design, (2) merged identical properties, (3) grouped the properties that complementarily define a higher-level one (e.g., physical coupling), (4) precisely defined the properties that we found unclear, and (5) structured them with the designer in mind (e.g., into macro and micro properties). We then (6) studied their dependencies.

We now present the space of properties resulting from this analysis. Through this literature review, we found that there is no unified design space for modular shape-changing UIs.

On the one hand, the robotics field contributed to the design space by conducting extensive surveys of previous implementations. From these surveys, they draw 1) classification and evaluation methods [1, 20, 30, 108, 123], 2) benefits and challenges of modular self-reconfigurable systems [8, 102, 122], 3) criteria for design [9, 11]. However, these contributions are not centered on the user. In particular, many of the publications presenting tools for design are centered on the technology. As we aimed for our properties to be independent from the technology in order to prevent obsolescence of the properties, we did not consider the properties that were centered on the technology. For instance, the docking mechanism (e.g., [44, 58, 66, 117]) is a low-level hardware implementation concern that is centered on the technology, and contributes to two of our user-centered properties: *strength* and *smoothness and resolution of the envelop*.

On the other hand, the HCI literature provides surveys and design tools centered on the user. Their contribution are: (1) classification and evaluation methods [47, 87, 103, 104], (2) benefits and challenges [2, 38, 84], (3) criteria for design [57, 63, 91], (4) how users perceive and interact with modular shape-changing UIs [51, 60, 74]. However, these tools mostly deal with the broader field of shape-changing UIs rather than focus on *modular* ones. Thus, they do not take into account the specificities of modular robotics nor the variety of robot configurations. E.g., a swarm and a chain implementations do not allow the same ranges of shape-change.

Some design rationales are proposed in papers presenting implementations. For instance, the authors of Zooids stress the importance of the size of modules, as they advocates for modules as small as possible in order to make tangible UIs made of "stuff" rather than "things" [57]. However, each design rationale consider a very local and specific set of designs, rather than provide a global viewpoint. For example, LineFORM [73] proposes a design space specifically for *actuated curve interfaces*. Cubimorph [91] specifically targets *handheld chain interfaces*. Zooids [57] defines requirements for *tabletop swarm interfaces*. In addition, our literature reviews reveals inconsistency between the design rationales. For instance, Zooids require their modules to be constantly detached, while Cubimorph require theirs to be constantly attached.

4 PROPERTIES OF THE INTERFACE (MACRO SCALE)

From the systematic literature review, we identified and structured the following set of 10 design properties that apply to modular shape-changing UIs (Figure 3 and summarised in Table 1). We classify them according to which level between the system and the user they impact most: *digital*, *physical* or *interaction*:

(1) *Digital level*: The properties at the digital level are related to the position and state of the modules in the computational model. Even though we left for future work the reconfiguration algorithms, we report here other high-level digital properties that impact user interaction.
(2) *Physical level*: The properties at the physical level are the ones of the tangible artifact made of physical robotic modules, i.e. the physics of the UI. They impact user interaction as users will interact with the physical UI, but are not considered as being at the *interaction level* since they can be characterized even if there is no interaction.
(3) *Interaction level*: The properties at the interaction level primarily impact the interaction design of the UI.

Physical and digital levels were introduced in prior work (e.g., [15]), while the interaction level stems from the current knowledge on how users interact with a modular shape-changing UI. We use ergonomics criteria [5] to characterize the expected impact of these properties on the user : they all (in)directly impact user experience as we adopt a user-centered approach, even though physical- and digital-level properties can be characterized out of an interaction context.

4.1 Coupling between modules (Digital /Physical)

Coupling between modules can be described both at the digital level (*hierarchy*), and at the physical level (*smoothness* and *resolution*).

Hierarchy defines whether the modules are standalones (i.e., driving the action by themselves) or satellites (i.e., the action is dependent on a standalone module or group of standalone modules) [27]. Satellites can be either original satellites (i.e., always synchronized to the same standalone module(s)) or borrowed satellites (i.e., able to synchronize to any standalone module(s)). Current approaches in robotics range from *standalone* (e.g., [27]) to *standalone + their satellites* (e.g., [53, 109]). In some cases, all modules are satellites and the device assuming the role of "standalone" is externalized (e.g., a micro-controller [57] or overhead controller [93]). However, it is not suitable for every context of use: e.g., a static overhead control system does not suit mobile use. Homogeneous systems do not inherently have to follow the standalone approach: modules may have the same hardware and software design but switch between satellite and standalone roles depending on the needs of the system or user.

Smoothness of the envelop is defined by deviations from the envelop, of the direction of the vector normal to the surface modules. We simplify this as the *gaps between modules*. A very smooth surface has been a requirement for decades, from the "operating surface texture" [21] in 1995 to the "seamless interactive surface" [91] in 2016.

Resolution of the envelop is defined as the input and output resolution at the surface of the UI, in dots per square centimeter ($\mathrm{d\,cm^{-2}}$) [88]. Researchers aim towards interfaces which provide high input resolution, and a high output resolution for expressive shape-change capacity [2].

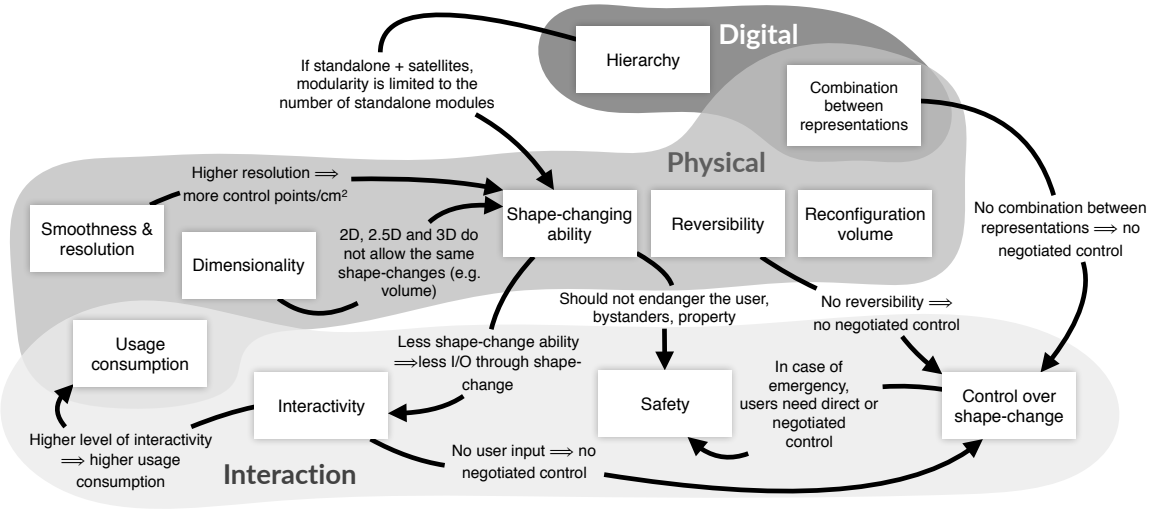


Fig. 3. The user-centered properties for modular shape-changing interfaces, and their dependencies. The properties are further classified depending on whether they impact the interface at a digital level (dark gray), physical level (middle gray) or interaction level (light gray). *Combination between physical and digital states* is classified as impacting both digital and physical levels, as it describes the interaction between the physical and digital shape of the interface. *Usage consumption* is classified as impacting both the physical and interaction levels as it is the power consumed *under a standard context of use*.

The approaches to achieve a smooth envelop with high resolution depends on whether we consider each module as 1) an *individual pixel/voxel/sensel* [88], or 2) as a *set of pixels/voxels/sensels*, i.e. a small display.

The first approach, where modules are individual pixels, can achieve both high *smoothness* and *resolution* through micro-scaled modules (e.g., [2, 9]). Although this approach offers more flexible and expressive shape-change, as each voxel/sensel is able to reconfigure in the space, current implementations are at the cm-scale rather than the desirable mm-scale [2]. Nonetheless, with current modules as small as ⌀11 mm [7], the micro-scaled approach is promising for long-term research as the size of components will decrease.

The second approach, where modules offer a small display, can achieve high *smoothness* through minimal joints between modules [91], and high *resolution* through, e.g., high resolution touchscreen displays on the faces of the modules. Even though current lattice systems are at the cm-scale, cubic implementations (e.g., [25, 26, 90]) provide seamless surfaces through minimal joint between modules. A few existing implementations embed displays on their modules ranging from arrays of LEDs (e.g., [72]) to OLED/FOLED screens (e.g., [27, 28, 91]). Swarms hardly allow for smooth physical coupling: few provide latching capabilities (e.g., [18, 68]) but most are constantly unlatched (e.g., [57, 93, 107]). This prevents the envelop of the interface to be smooth.

4.2 Combination between the digital and physical states of the UI (Digital / Physical)

A modular shape-changing UI is characterized through two states: (1) its physical state and (2) its digital state, also referred to as "computational representation" [9]. The physical state can be described as the tangible object made of physical modules. The digital state can be described as the position and state of the modules in the computational model, due to the modules having to compute collectively in order to achieve a common goal. For perspective, non shape-changing TUIs only have a single physical state, i.e. a single physical shape [99].

If the digital state changes, the physical state should change in accordance and vice versa. E.g., if the physical state changes through direct user shape deformation, the digital state should change to match the new physical state [9]. With no combination between physical and digital states, the usability of the system may suffer from inconsistencies between the physical and digital states.

Deterministic reconfiguration better allows the combination of physical and digital states, as the system knows the position of each module at all time. Stochastic implementations, however, hardly allow for combination between states: the system only knows the position of a module when it comes in contact with another, and the paths taken by the moving modules are unknown [122]. Thus, even though the computational and physical representations do match *before* and *after* the shape-change, the stochastic nature of the reconfiguration does not allow the computational representation to match the physical representation *during* the change of shape.

4.3 Shape-change ability (Physical)

Shape-change ability quantifies how much the UI can change its shape. We characterize the shape-change ability of a system through the 11 Morphees+ features [47], which provide 11 quantifiable features drawn from two established taxonomies. For instance, drone-based swarm UIs (e.g., [10]) allow limited *Curvature*s [47] as drones cannot stack. The physical shape of the interface should be able to vary over time [9, 38], in order to provide input and/or output modalities [2, 38, 87], and to support the "form follows flow" design principle [34].

We will discuss in particular the *Porosity* and *Modularity* features [47], as modular systems uniquely enable them. First, *Porosity* is the ratio of the area of perforated parts to the total area of the shape [47]. Modular systems uniquely enable a large range of sizes for holes.

Second, *Modularity* is the ability of an interface to be split in at least two parts and (re)combined while maintaining its original functionality. It is computed as the number of functionally possible combinations. [47]. Prior work [91] proposes to provide users with permanently attached modules so that they cannot fall off or be lost. However, this hinders the *Modularity* of the interface. Chain systems do not allow for changes in *Modularity* as they either do not support detach-ability (e.g., [67, 73, 94]), or in a limited way [72, 91, 100]. For instance, ChainForm [72] users can unlatch modules from a neighbor, but the unlatched modules become unfunctionnal. Lattice systems can be fully detach-able and allow for *Modularity*: a module can unlatch from all its current neighbour at once (e.g., [9, 90, 127]). Lattice systems can also be partially detach-able and do not allow for *Modularity*: a module needs to first latch to a new neighbour before unlatching from its current neighbour (e.g., [76]). Swarm systems are either detachable or permanently detached. Both cases allow for *Modularity*. Although permanently detached modules cannot be "physically" split through latching/unlatching, they can be "functionally" split (e.g., [50]) with two subsets of modules being their own interface, and merging back into one. We consider this behavior to match the defintion of*Modularity*.

4.4 Reversibility (Physical)

Reversibility defines if the system is able to return to its initial state and repeat the shape-change. Shape-changing UIs should allow for reversibility [87]. Deterministic systems using self-reconfiguration usually allow for reversibility. A design flaw that may impair reversibility is if the system allows modules to be only added or only removed but not the other way around. Thus, implementations solely based on self-assembly (e.g., [35]) or self-disassembly (e.g., [25, 26]), and stochastic systems, only *partially* allow reversibility: if the user wants to return to a previous shape, they need to repeat the whole self-(dis)assembly process. This leads to usability problems: as stated by the *Minimal action* ergonomic criterion [5], the number of actions the user has to perform should be minimized. Requiring users to repeat the whole process adds extra steps and increase their workload. Similarly, if the user makes a mistake, they should be able to correct only their previous action(s) and not have to repeat the whole process (*Error correction* criterion [5]).

4.5 Dimensionality (Physical)

Existing shape-changing interfaces are able to reconfigure either in 2D (e.g., [4, 57, 72]), 2.5D (e.g., [107]) or 3D (e.g., [43, 90, 91]). Although we found several instances of 2D and 3D working prototypes, 2.5D implementations are seldom explored. Pin-based shape-changing interfaces (e.g., [22]) may provide insight to design future 2.5D modular shape-changing UIs.

The majority of swarm implementations only allow for 2D shapes on a flat surface (e.g., [57, 93]). Some provide 2.5D reconfiguration (e.g., [107]). 3D reconfiguration is supported by drone-based swarm UIs (e.g., [10]). Although chain implementations mostly allow for 3D reconfiguration, some only reconfigure in 2D (e.g., [46, 96, 125]). Notably, ChainForm's modules [72] are only capable of 2D planar transformation. They require the user to manually add a plastic joint between two modules so that each reconfigure in different 2D planes. Thus, the system can further achieve 3D transformations, albeit limited.

Similarly to chain implementations, most lattice implementations allow for 3D reconfiguration with few exceptions (e.g., [12, 77, 113]). Current working stochastic systems only allow for *2D reconfiguration*. Solutions to achieve stochastic reconfiguration in 3D have been explored, with the example of a shaken bag containing the modules [25] but only consist of concept scenarii or simulations.

4.6 Volume required for shape-change (Physical)

The volume required for shape-change is the total volume occupied by the system throughout its change from the initial to the target shape. The total space used for reconfiguration should not exceed the union of the initial and target shapes [109]. Doing so, a user can anticipate the room required for a change of shape, even if actuated by the system [38, 91]. For instance, the device may be held in a single hand or placed on a table. In these cases, the interface should not fall or bother the persons around the users. This property is an issue for stochastic implementations where the modules can, and must, move in a large space in order to maximize their chance to encounter a relevant neighbor.

4.7 Usage consumption (Physical/Interaction)

The usage consumption is the power consumed under a standard context of use and the resulting expected duration of the interaction. When buying state-of-the-art devices (e.g., tablets, smartphones, laptops), the technical specifications describe not only the battery model, but also the expected autonomy under standard usage (e.g., *"Up to 10 hours of surfing the web on Wi-Fi, watching video, or listening to music"* [3]). Zooids [57] can move for one hour and can work longer under normal usage (i.e., where constant movement is not required). The usage consumption of shape-changing interfaces is little studied in the literature, with only few papers mentioning it, although it was flagged as a grand challenge of shape-changing interfaces [2, 14].

A challenge lies in the lack of knowledge of the standard usage of modular shape-changing UIs. Evaluating standard usage is all the more difficult since 1) modular shape-changing UIs are still constrained to research prototypes, and 2) current prototypes are often not robust enough for longitudinal evaluation [2, 10]. As a result, previous work proposed concept scenarii (e.g., [10, 57, 91]) and most user experiments were conducted with 2D UIs (e.g., [52, 105]). A reasonable minimum threshold to enable user studies would be one hour. For mobile and wearable shape-changing interfaces, previous work set the goal to a full day [2].

4.8 Control over shape-change (Interaction)

Control over shape-change describes who controls the change of shape and how this control is shared between the user and the system. Existing shape-changing interfaces provide different levels of control over shape-change [86]: direct, negotiated, indirect or system control.

Direct control (i.e., full user actuation) allows for the shape-change to be solely driven by the user, through direct shape deformation (e.g., [27, 53, 64]). *Negotiated control* between the user and the system allows for both the user and the system to initiate and further share the control over the change of shape (e.g., [57, 72]). *Indirect control* allows the system to initiate and further control the shape-change, based on inferences or interpretations of the actions of the user (e.g., [17]). As user control is "implicit", users need to understand the modality that the system uses to infer/interpret user action if they want to knowingly control the shape-change. E.g., Actuating Mood [17], although non-modular, changes shape according to user emotion: the user could "force" the system into their desired shape by mimicking the required emotion. *System control* (i.e., full system actuation) allows the system to solely initiate and further control the change of shape, without any user input, neither direct nor indirect (e.g., [7, 91, 93]).

While the HCI community has been stressing the importance of actuation [38, 99], some modular systems only allow for users' direct control of the shape (e.g., [27, 53, 64]). This is in line with the *Explicit user action* criterion [5]: the system should only react to *explicit* user action. The system should anticipate every possible user action and provide appropriate options to keep the user in control of the interaction (e.g., interrupt, pause or continue the shape-change, come back to the previous step) (*User control* criterion [5]). Moreover, the higher the level of control the user has on the shape-change, the higher their level of trust *but* the higher their workload [74]. In the future, very small modules will also cause direct control to become unpractical. Thus, users should only be required to do the least number of actions necessary to accomplish a task, for a minimal workload (*Minimal action* criterion [5]). Actuated shape-change can help minimizing the number of steps to go from an initial shape to a target shape.

Negotiated control allows for the user to explicitly act towards shape-change, and for this shape-change to be computationally controlled. Swarm systems from HCI provide negotiated actuation between the system and the user (e.g., [10, 57]). Actuation of current chain systems are either system controlled or negotiated. Deterministic reconfiguration allows all types of control on shape-change, from directly controlled by the user to fully controlled by the system, whereas all of current stochastic implementations currently only allow for system control.

4.9 Interactivity (Interaction)

Interactivity describes the modalities the system offers for the user to interact with it. The system can support user interaction, at least through shape-change and possibly through further input/output modalities [2, 57, 91]. Based on the related work, we propose to distinguish the following dimensions for interactivity:

(1) *Changes of shape* can provide *input and/or output*, e.g., input through direct shape deformation and output through visual or haptic feedback [52, 73].
(2) *Other modalities* can provide *input and/or output*, e.g., on each modules, direct touch interaction for input and a colored "pixel" LEDs for output [10, 57] .

The most interactive systems come from the HCI field, implementing various modalities for both input (e.g., touch [57, 91], mid-air gesture [10, 101], direct shape deformation [73, 106], wearable glove-like controller [114]) and output (e.g., LEDs [10, 57, 72], GUI displays [28], vibrotactile feedback [114], haptic patterns [52]). However, even though systems coming from the robotics field rather focus on locomotion, and that reconfiguration is not designed to *interact with users*, prior robotics work embed sensors to *interact with their environment* (e.g., temperature sensor [29], ambient light sensor [93]).

An important research challenge for modular shape-changing UIs is to lead the user to perform relevant actions (affordance), and inform the user on the alternatives when several actions are available (*Prompting* ergonomic criterion [5]). It is knowingly difficult to inform users of the transformational capabilities of a shape-changing UI [34, 38, 112]. This is even more difficult for modular shape-changing UIs as modular robots provide large transformational capabilities, unlike other shape-changing UIs (e.g., [49]). Previous work encourage designers to

(1) explore affordances through shape-change [118], interaction design and material design [63] and (2) to study how users perceive and interact with modular shape-changing UIs [50].

The currently limited *shape-change ability* and *interactivity* of working implementations challenge the design of meaningful and usable interaction techniques. If the system only implements shape-change as output, the challenges lie in the design of *dynamic affordances* [38, 61, 112] with the currently limited *shape-change ability*. If the system implements other output modalities, e.g., graphical displays [10], sound [53] or kinesthetic feedback [72], designers can leverage them for *prompting*. An alternative is to delegate the prompting to an additional device. For example, designers explored prompting through a separate television monitor [50], through wearables providing vibrotactile feedback [114], or through AR or VR, e.g., for data physicalization with swarms [33, 107].

4.10 Safety (Interaction)

Safety quantifies how much the system may endanger the user [91], bystanders or cause damage to physical property [2]. The reconfiguration and final shape of the interface should not cause danger. A risk should be assessed through its severity, the possibility of avoidance and the redundancy [40, 41].

Severity is the pressure applied on a body part. The smallest maximum force that can be applied on a user's body part is against the face, with 65 N (pressure of 110 $\mathrm{N\,cm^{-2}}$) [40, 41]. This sounds therefore like a limit for the UI maximum force in case the interface hits a face. However, as the global pressure applied on the body depends on the contact area, the force can be high if the part of the UI touching the user is sharp or small. Solutions include designing non-sharp shape-changing UIs or using soft materials.

Possibility of avoidance is inversely proportional to the speed of the system. To enable avoidance, solutions include low speed for shape-change, embedding proximity sensors for stopping the reconfiguration when approaching the user –although this may hinder *interactivity*– or including an emergency stop button. A design challenge lies in finding an acceptable balance between low speed for safety and high speed for immediate feedback [48] (ergonomics criteria [5]).

Redundancy is the amount of time the risk is possible over a fixed amount of time.

Despite its importance, safety is seldom studied in the HCI literature (a rare example is found in [91]). Robotics, due to decade of research on actuated physical movement, has thoroughly studied safety.

Table 1 summaries all the properties of the interface at the macro-scale, from the *coupling between modules* to the *safety*, together with their possible values and examples from the literature. This table can serve as a basis for comparing and designing modular shape-changing UIs.

5 PROPERTIES OF THE MODULES (MICRO SCALE)

In this section, we present the properties at the micro-scale of the modules (Figure 4). The following micro-properties are technical: we will further discuss in Section 6 their impact on the user experience through their impact on the macro-scale properties.

We classify them according to their abstraction level: properties at the *intra-module* level characterize elements that are embedded in the module. Properties at the *module* level characterize the envelop of the modules. Properties at the *inter-module* level characterize if and how modules are attached to each other.

5.1 Input, output and processing capabilities (Intra-module)

Input, output and processing capabilities define how users interact with a *single* module. Input and output capabilities are dependent on micro-devices embedded in the module. For example, input can be supported by touch sensors [57], microphones [53], or ambient light sensor [93]. Output can be supported by a LED [72], a graphical display [28], or a speaker [53]. Processing capabilities depends on the microprocessor.

Properties		Values		Examples
Coupling between modules	Hierarchy	Standalone | Original satellites | Borrowed satellites		Standalones [114] Original Satellites [69] Borrowed satellites [108]
	Surface smoothness	Size of gaps (mm)		~0 mm [90]
	Resolution of the envelop	Number of dots per unit area (d.cm^{-2})		0.69 d.cm^{-2} [23]
Combination b/w physical and digital states		Yes | No		Yes [55, 89, 90] No [10]
Shape-change ability		Morphees+ features [46]		Modularity [55]
Reversibility		Yes | No		Yes [90] No [23, 24, 33]
Dimensionality		2D | 2.5D | 3D		2D [4, 55, 69] 2.5D [106] 3D [7, 89, 90]
Volume required for shape-change		Volume (cm^3)		381(L) × 381(W) × 100(H) mm^3 [21]
Usage consumption		Time under normal usage (min)		60 min while moving [55]
Control over shape-change		Direct | Negociated | Indirect | System		Direct [51] Negotiated [55, 69] Indirect (none) System [7, 90, 92]
Interactivity		Input	Shape-change | Other	Shape-change [51, 70] Other [55, 92]
		Output	Shape-change | Other	Shape-change [88, 105] Other [10, 69]
Safety	Severity	Pressure on skin (N.cm^{-2})		Not mentioned
	Possibility of avoidance	Inversely proportional to speed (m.s^{-1})		0.5 m.s^{-1} [55] 13 cm.s^{-1} [99]
	Redundancy	Probability over fixed amount of time		Not mentioned

Table 1. Properties of the interface (macro-scale) for modular shape-changing UIs.

5.2 Power storage (Intra-module)

Power storage defines how the modules are powered. Systems are either battery-powered (e.g., [90, 93]) or mains-operated (e.g., [73]). Battery-powered systems embed batteries in each module. Charging the modules is a challenge [57]. The most common approach is to charge each module individually (e.g., [57, 89]), which can be tedious as the number of modules increases. Other approaches propose the use of a charging dock [93], or a photo-voltaic cell embedded in each module [124].

Mains-operated systems are either (1) heterogeneous, i.e. one [72] or few [53] modules are plugged-in and provide power to the others, (2) homogeneous, i.e. all modules are plugged-in [12], or (3) externalized, i.e. the modules themselves are not plugged-in, but the environment in which the reconfiguration takes place is (e.g., for stochastic [32] implementations). The heterogeneous approach is hardly suitable for large scale systems, as the modules furthest from the power supply should not see a significant drop in voltage [124]. Moreover, both heterogeneous and homogeneous modules have the issue of the power chord which needs to be taken into account during reconfiguration and/or user interaction, as it can get in the way.

5.3 Size (Module)

The size is the volume of each module, defined by the length of the edge for cubic robots [90] or the diameter for spherical robots [82]. We find many sizes in the literature: the M-Blocks' edge measures 50 mm [90], while

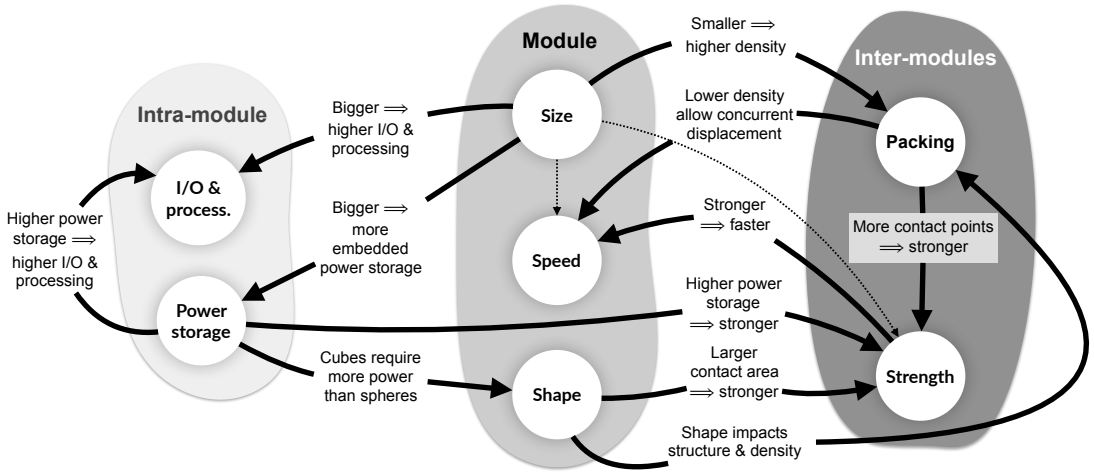


Fig. 4. The properties of modular shape-changing UIs at the scale of the modules, and their dependencies. The properties are further classified depending on whether they impact the components embedded in the module (intra-module, light gray), the whole module itself (module, middle gray) or at least two modules (inter-module, dark gray). The dotted arrows describe yet undefined impact requiring further technical experiments.

Catoms measure ⌀11 mm [9]. As often with electronics, their size is getting smaller as technology advances. The size of modules was presented as a central property in order for UIs to be made of "stuff" rather than "things" [57]. The hardware required for the self-actuation of deterministic systems limits our ability to manufactured the modules at a very small scale and in a large amount [119].

5.4 Shape (Module)

The shape defines the geometry of a module. While the most widespread shapes are cubes (e.g., [25, 90, 98]) and quasi-spheres (e.g., [9, 76, 100]), many other geometries are possible, such as rectangles [72], x-shaped [4, 55], or cylinders [57, 93].

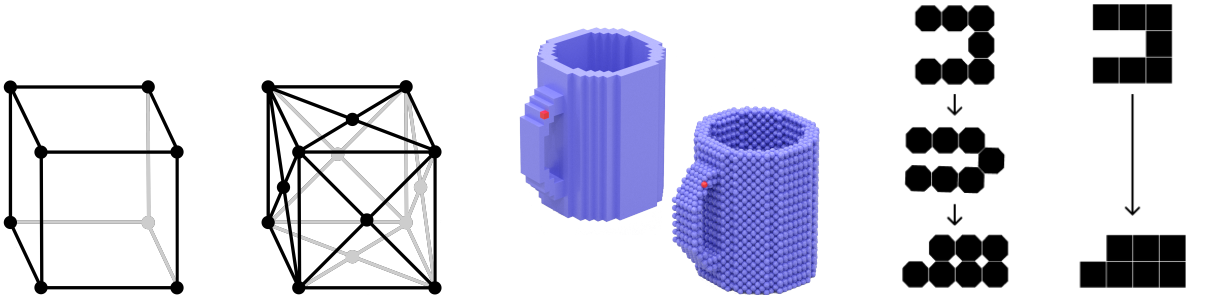
5.5 Speed (Module)

The speed defines how fast a single module can move. For electrostatic or electromagnetic actuation (e.g., [82, 90]), the speed is driven by the distance between two motion actuator electrodes. Designers should take into account the trade-off between the latency to trigger the latching and the distance the module covers. E.g., if the distance between two motion actuators is small, the latching will be fast but the module will cover little distance. For mechanically-actuated implementations, the speed depends on the force of the motors. The triggering latency of these implementations is higher than for electrostatic/electromagnetic actuation, as they require more steps (e.g., align the latching mechanisms, latch, check the latching, move).

5.6 Packing (Inter-modules)

The packing defines how several modules can be spatially arranged together to form the interface. Packing can be divided into two parameters: *structure* and *density*.

Structure defines the way of arranging modules so that they cover a volume without overlapping. When looking for a complete coverage of the volume, i.e. with no holes between the modules, the mathematical problem is



(a) Simple cubic lattice.

(b) Face centered cubic lattice.

(c) Quasi-spherical robots better approximate curves than cubes.

(d) Quasi-spherical robots achieve higher closure resolution than cubic ones.

Fig. 5. (a–b) Two possible lattices for spatial arrangements (packing) of robots, (c) approximation of a shape by cubic and quasi-spherical robots, and (d) range of *Closure* achieved by cubic and quasi-spherical robots.

known as "tesselation" or "honeycomb" [31]. When a complete coverage is not necessary, lattice and swarm systems can follow a crystal structure [36] which describes the possible arrangements of fixed-shaped elements. For instance, one can use a Simple Cubic lattice (Figure 5a) or a Face-centered cubic lattice (Figure 5b). However, chain structures usually aim for locomotion rather than coverage. For example, common chain structures are caterpillars, wheels/crawler and multi-legged walkers [56].

Density defines if the structure is fully filled (high density) or not (lower density). The density is described by its number of units per volume, and applies to all types of systems.

5.7 Strength (Inter-modules)

The strength defines the force of the connection between a module and its neighbor(s). Strength depends on the strength of the actuation mechanism. E.g., the force of the motors for mechanical actuators (e.g., pins [107], hinges [91], clamps [76]) and the force between two conducting electrodes for electrostatic or electromagnetic actuators (e.g., [82, 90]). Strength is zero for swarm modules, as they cannot attach to each other.

5.8 Others

Other parameters exist at the module level, and include the weight and coating of modules, as well as the noise or heat they generate. We found that the weight is often correlated with size, except for *"helium catoms"* [45] which are 8 m^3 modules filled with helium, and thus big but light by design.

6 IMPACT BETWEEN PROPERTIES

We discussed the properties at the macro-scale of the interface (section 4) and at the micro-scale of the modules (section 5). However, we need to further take into account the dependencies between them. Not only do macro-scale properties impact each others, but micro-scale properties impact the system at both the micro- *and* macro-scales. Thus, technical properties of the modules may indirectly impact user interaction with the interface through their impact on macro-scale properties.

6.1 Impact between macro-scale properties

Figure 3 shows the dependencies between macro-scale properties. We present them in decreasing number of dependencies: *shape-changing ability* impacts or is impacted by 5 other properties, while *control over shape-change* impacts or is impacted by 4 other properties, and *interactivity* impacts or is impacted by 1 other property.

Shape-changing ability ←[1] *Dimensionality.* Systems able to reconfigure in 2D, 2.5D and 3D do not allow for the same changes in shape: e.g., 2D implementations do not allow for changes in volume, while 2.5D implementations do not allow for changes in closure and their change in porosity is limited as they do not allow for bridges. Only 3D implementations were found to allow for all shape-change features.

Shape-changing ability ← *Hierarchy.* The ability to change modularity is dependent on the number of standalone modules and the nature of its satellites, if any. The *Modularity* feature from [47] is described as the ability of the interface to split *and maintain its original function*. This causes no issue with the standalones approach, as each module can dissociate and keep functioning normally. However, when taking the standalones+satellites approach: (1) If the satellites are original satellites, removing a standalone means removing all of its synchronized satellites. If some remain in the original interface, they will stop functioning. (2) If the satellites are borrowed satellites, removing a standalone means removing none, part or all of its synchronized satellites. If some remain in the original interface, another standalone has to be available for them to synchronize and continue functioning. (3) If the standalone is externalized (e.g., a micro-controller [57] or overhead controller [93]), it cannot be removed from the interface. The satellites allow for *Modularity*.

Shape-changing ability → *Interactivity.* The higher the shape-change ability, the higher the input/output possibilities through shape-change. The way shape-change conveys users input and/or system output depends on its shape-change ability. For example, ChainFORM [72] proposes a shape-changing stylus interface able to switch between a pen mode, a brush mode and a magnifying glass mode. This is enabled by the capability of ChainFORM to change *Closure*. An implementation which does not allow for *Closure* (e.g., ShapeBots [107]) would not enable this scenario. The input/output of a system and its required shape-change features will depend on the context of use. For example, permanently detached swarms do not allow users to grab a part of the interface like a solid object, as the UI is made of detached modules. Thus, although permanently detached swarm systems allow for great modularity, they may not be suitable depending on the context of use (e.g., handheld). However, a system with high shape-changing ability will overall support more input/output possibilities through shape-change.

Shape-changing ability → *Safety.* The shape taken by the interface before, after and during reconfiguration should not endanger users, bystanders or properties. Shape-changing features that may be a source of concern are the *Speed* (e.g., participants expressed fear of getting hurt because of the quick reconfiguration of KnobSlider [48]), *Curvature* (e.g., if the interface is able to achieve really sharp edges) and *Strength* (e.g., with wearable UIs [2, 72, 73], very strong shapes could harm the bones or joints).

Control over shape-change → *Safety.* The user should always be in control of the interaction, and particularly able to interrupt, cancel, pause and continue the reconfiguration (*User Control* ergonomic criterion [5]). This applies particularly to safety: when the user detects a safety concern, they should be able to stop the reconfiguration, e.g., if an on-body implementation (e.g., [73]) is hurting them or a drone (e.g., [10]) is flying toward a bystander. Explicit user control ensuring safety requires direct or negotiated control.

Control over shape-change ← *Interactivity.* The system should support user input to allow explicit user control. Systems that do not support user input, either through shape-change or any other way (e.g.,[43, 89]) do not allow for direct and negotiated control.

Control over shape-change ← *Combination between physical and digital states.* Negotiated control requires combined physical and digital states. If there is limited combination between both states (e.g., the physical states changes according to the digital state, but not the other way around), and the user changes the shape through

[1] "depends on" or "is impact by"

direct deformation, the system will not be able to match its known computational representation and the new positions of the modules in the physical space. Thus, the system will not be able to move the modules, and only direct control will further be possible.

Interactivity → Usage consumption. We expect that the higher the interactivity, the higher the usage consumption. For example, if a UI allows user input through shape-change, its usage consumption will increase as modules need to invest energy in listening to a potential modification of their location and communicating the new location to their neighbors. If a UI allows input and/or output through other modalities, its usage consumption will increase as the hardware to enable these additional modalities need to be powered.

6.2 Impact between micro-scale properties

The properties of the modules are not independent but impact each other. Designers should carefully consider the dependencies between properties when deciding on a design. Changing one property may not only impact another one directly (e.g., the size impacts the strength), but also indirectly (e.g., the size of the modules impacts their packing which in turn impacts the speed). We now present the property having the most impact, i.e. the *size* of the modules.

Size ↔ Input/output and processing capabilities. The bigger the modules, the higher the I/O and processing capabilities. Reducing the size of the modules implies reducing the space available to embed components. As a consequence, the smaller the modules, the smaller or the fewer the components. For example, on the one hand, the smallest existing implementations (11mm [9] and 12mm [25]) do not embed any component supporting input, and output is supported solely through shape-change. On the other hand, the ⌀26mm Zooids [57] are the smallest implementation embedding components for both input (touch sensor) and output (colored LED) in addition to shape-change. The more the memory of the microprocessor, the larger the surface it requires.

Size → Power storage. Smaller modules imply smaller *embedded* power storage. For battery-powered modules, smaller batteries means less power storage. For example, the Michigan Micro Mote (M3) [78], a mm-scale computing system, use 0.5–5 Ah batteries. For comparison, the cm-scale M-blocks [89] embed 4×125mAh batteries. The size of the components is less of an issue for mains-operated implementations, as the components to power the modules are smaller than batteries (e.g., wires, connectors between modules). The smaller the modules the lighter, hence they need less power to move.

Size → Packing. The smaller the modules, the higher the density. For a same structure, decreasing the size of the modules results in more units per volume. However, designer should also take into account that the smaller the modules, the more the required fabrication accuracy to enable accurate packing: accurate packing of 5.0 mm ± 0.5 mm modules is easier than the packing of 1.0 mm ± 0.5 mm modules.

Size → Speed. Further technical experiments are needed to establish the link between size and speed, taking into account the actuation mechanism, the weight and distance between two modules. First, the smaller the modules, the slower the reconfiguration of the shape, as there are more modules to move. Second, the smaller the modules the lighter, hence the faster their displacement for the same power. Third, the smaller the modules, the smaller the distance to cover to move a module. The smaller the distance, the faster triggering of an electrostatic-/electromagnetic-actuation. E.g., magnets attract each other quickly if they are close. However, for electrostatic-/electromagnetic-actuation, a small distance between two motion actuators also implies a small the distance covered by a module. As a consequence, there is a trade-off between the triggering latency (i.e., how fast the latching occurs) and the effective distance covered.

Size → Strength. On the one hand, decreasing the size of the modules does not significantly impact their *individual strength*. The reason for this is that the modules are light when small, and consequently require less power for actuation [82]. On the other hand, the strength of the UI that the user will manipulate is the *strength*

per unit volume. Smaller modules lead to a higher density of modules per unit volume, which may feel stronger in the users' hands.

Strength ← Shape. With electrostatic and electromagnetic actuation, the larger the contact area, the stronger the bond between modules [82]. E.g., for a same size, a cubic module has bigger faces than a quasi-spherical module. For mechanical actuators, we could not find an impact of the shape on the strength.

Strength ← Packing. The higher the density, the higher the strength per unit volume. A way of increasing the strength is to increase the number of neighbors for each module (i.e. density).

Strength → Speed. The higher the strength, the higher the speed. For electrostatic and electromagnetic actuation, the speed depends on the force of the bond between two actuator electrodes: the stronger the connection between two modules, the faster they can move – as they can provide more force to latch and attract a neighbor [82].

Strength ← Power storage. The higher the power storage, the stronger the connection. When the connection between modules needs power, the more power storage, the stronger the connection between two modules. Electrostatic actuation needs high voltage for strength [82], while electromagnetic and mechanical actuation needs high amperage. Both can be provided by a high power storage.

Packing ← Shape. The shape of the modules impacts their packing [81]. All shapes do not allow for the same number of actuators, nor for the same disposition of neighbors around a module. E.g., a module made of two half-cylinders joined on their round side can accommodate two neighbors [43], whereas cubes can have six [89]. Quasi-spherical designs can also have up to six neighbor [82]. However, where the position of the six neighbors on a cube is constrained to its six faces, quasi-spheres allow more possible placements (e.g., Figure 5d). Hence, cubes and quasi-spheres allow the highest density, but quasi-spheres allow the most diverse structures.

Power storage → I/O and processing capabilities. The higher the power storage, the higher the I/O and processing capabilities, i.e. the more components the module will be capable to power. Similarly, processing requires power.

Power storage ← Shape. Cubic shapes require a lot of power in order to move the robots [82]. Latching and moving cubes around each other, while staying connected, is hard to implement in miniature, cubic, versions. Quasi-spherical shapes enables latching and moving with lower power.

6.3 Impact between micro- and macro-scale properties

In this section, we discuss how the technical, micro-properties of the modules impact the user-centered, macro-properties of the interface. Our goal is to inform researchers on how the technical choices of their design can impact directly and indirectly the usability of the whole interface. We present these relationships starting from the macro-property with the highest number of impacted properties to the least.

Shape-change ability is impacted by six micro-scale properties:

Shape. The shape of the modules impacts the ability to change the *curvature*, *closure*, *zero-crossing* and *amplitude* (Figure 5d). Current flat, rectangular interfaces are better formed by cubic robots. However, non-rectangular, non-flat interfaces are promising: e.g., users can be more efficient when pointing on non-flat touchscreen [92].

Size. The smaller the modules, the more the actuation points per cm. Thus, the size impacts the *granularity*, *curvature* and *porosity*.

Packing. Being able to change the structure and the density impacts many shape-change features. First, if the object is not very dense, the movements of several modules can be concurrent [109] for a faster *speed* of shape-change. Second, lowering the density of the inner structure by moving inner modules to enlarge the envelop will allow for changing the *size*. Lowering the density and choosing a specific structure could also allow for a change in the *stretch-ability* of the interface [42]. Third, denser packing allow for more control points per unit area, hence a higher *granularity*. Fourth, a change in *porosity* correlates with a change in density. The higher porosity, the fewer the units per volume, hence the lower density.

Micro properties	Macro properties	Impact micro-macro (summary)
I/O & Processing	Hierarchy	Standalones require more processing capabilities
	Combination between states	Higher I/O & processing ⟹ better combination
	Control over shape-change	More I/O & processing ⟹ higher control
	Interactivity	More I/O & processing ⟹ more interaction modalities
Power storage	Hierarchy	Standalone need higher autonomy
	Shape-change ability	Mains-operated systems impair shape-change
	Usage consumption	Higher power storage ⟹ higher autonomy
Size	Shape-change ability	Size impacts granularity, porosity, curvature
	Smoothness and resolution	Smaller ⟹ Higher resolution (individual pixel/voxel/sensel)
Shape	Shape-change ability	Gaps and bumps impact smoothness
	Dimensionality	Dimensionality depends on the shape (e.g., cylinders = 2D)
	Smoothness and resolution	Shape of modules impacts shape-change
	Safety	Sharp edges may hurt users
Speed	Shape-change ability	Fast speed ⟹ fast shape-change
	Control over shape-change	Speed-control trade-off
	Safety	Fast speed ⟹ little possibility of avoidance
	Interactivity	Speed must support immediate feedback
Packing	Shape-change ability	Packing impacts speed, stretch-ability, porosity
	Smoothness and resolution	High density ⟹ high smoothness & resolution
Strength	Shape-change ability	Too strong impairs modularity Too weak impairs curvature, closure and zero-crossing
	Safety	Strong connection ⟹ high severity
	Interactivity	Too weak/strong impairs direct deformation

Table 2. Expected impact of the properties at the scale of the modules (micro) on the ones at the scale of the interface (macro).

Strength. A strong connection impairs *modularity*. A weak connection impairs *curvature*, *closure* and *zero-crossing*. If the modules stick too strongly together, it will be difficult for users to split the UI (*modularity*). However, if the modules do not stick strongly enough, it will impact their overhang capabilities and prevent large ranges of *curvature*, *closure* and *zero-crossing*.

Speed. The faster the modules, the faster the reconfiguration of the whole interface. The *speed* at the micro-scale directly impacts the *speed* feature of the *shape-changing ability* property at the macro-scale.

Power storage. Heterogeneous mains-operated systems[2] impair the *modularity* of the interface: when splitting the interface in two, if the system only has one plugged-in module acting as a power supply for the others, the removed modules cannot keep functioning (*Modularity = 0*). If the interface has several plugged-in modules, these should be distributed between both halves of the split interface to power unplugged-modules and maintain functionality. Moreover, as the modules furthest from the power supply should not see a significant drop in voltage, the possible positions of the plugged-in module(s) in the global configuration are constrained. Another point of concern is the power chord, that should not get in the way during reconfiguration and interaction.

[2]In heterogeneous mains-operated systems, one [72] or few [53] modules are plugged-in and provide power to the others.

Interactivity, Smoothness, Resolution and Safety are each impacted by three micro-scale properties.

Interactivity is impacted by:

Speed. Speed must enable immediate feedback, i.e. usable output through shape-change [5, 57].

I/O & processing capabilities. The more the I/O & processing capabilities, the more the interaction modalities available.

Strength. Too high or too weak strength impairs direct deformation. On the one hand, if the modules do not stick strongly enough together, users would break the interface in thousands of pieces when they manipulate it. It would render input through shape-deformation challenging, and augment user's mental workload if they need to be careful when they manipulate the system. On the other hand, if the modules stick too strongly together, users may not be able to control the system through the change of shape. The research community should either find a compromise for the strength, or, even better, enable programmable strength.

Smoothness and Resolution are impacted by:

Size. The smaller the modules, the higher the resolution when modules are individual pixel/voxel/sensel. The size does not *directly* impact the resolution when each module embeds several pixel/voxel/sensel, i.e., a display. In this case, the resolution depends on the graphical display on the face of the modules (e.g., LED array [72], OLED display [91], FOLED display [28]). The size however *indirectly* impacts the resolution, e.g., through the link between size and I/O & processing capabilities (embedding an OLED display requires more space than LEDs), or size and power storage (an OLED display needs more power than LEDs).

Shape. On the one hand, cubic modules enable flat and seamless displays (e.g., Cubimorph [91]). On the other hand, spherical modules are better to approximate curves (Figure 5c), but result in irregular surfaces, with gaps and bumps (e.g., Figure 5c). However, this impact of shape decreases when the size of the module decreases: The smaller the robots, the less the shape of each robot impacts the tactile and visual perception of the objects [13].

Packing. The denser the packing, the higher the smoothness and resolution. On the one hand, high coverage (e.g., with using tessellation or honeycomb structures) and dense packing result in a high number of dots per centimeter. On the other hand, if the user only interacts with the *envelop* of the interface, the inner structure may not need to provide complete coverage, and may follow a less dense packing.

Safety is impacted by:

Speed. The speed of the modules may pose safety concerns. This risk should especially be taken into account if the modules have strong actuators (*Strength* property) and/or sharp edges (*Shape* property). E.g., if the user touches two modules with sharp edges that quickly and strongly latch, the skin can be pinched.

Shape. Sharp edges can be dangerous. For example, gripping an object made of cubes may hurt more because of the cube edges, rather than a smooth object made of spheres (e.g., Figure 5c).

Strength. The strength of connection between modules can be a safety concern, as they could apply too much pressure on the skin/limbs of the user. E.g., even though the users were afraid that the fast reconfiguration of the chained prisms of the KnobSlider [48] could hurt them, the servo motors of the device are not strong enough to actually cause any harm if their fingers get pinched. Strength should especially be taken into account with on-body implementations (e.g., [72]) where the interface applies forces against the users' body joints.

Control over shape-change and Hierarchy are each impacted by two micro-properties.

Control over shape-change is impacted by:

Speed. There is a trade-off between speed and control over shape-change. If the modules are too fast, the user may not be able to react fast enough. For example, the maximum speed of Zooids is 74 cm/s but their application use a slower speed (44 cm/s) for the user to better control the interface [57]. However, if the modules are too slow, this will impair immediate feedback [5].

I/O & processing capabilities. The more the input/output & processing capabilities, the higher the control over shape-change. In the literature, the smallest implementations do not include input modalities and are currently system controlled.

Hierarchy is impacted by:

I/O & processing capabilities. High processing capabilities allow modules to be standalones. To enable shape-changing modular UIs with modules embedding lower processing capabilities, researchers use an externalized system with higher processing capabilities as a "standalone" (e.g., microcontroller [73], overhead controller [93]).

Power storage. For battery-operated systems, the standalone(s) should not run out of battery. This would make them cease functioning and let their satellites alone, in which case the UI cannot function as expected. This can be an issue as the higher processing capabilities of standalones require more power.

Lastly, Combination between states, Dimensionality and Usage consumption are each impacted by one micro-scale property:

Combination between states ← *I/O & processing capabilities.* The higher the I/O & processing capabilities, the better the combination between the physical and digital states. When their physical representation changes, modules need more processing capabilities (e.g., to communicate with their neighbors to estimate their new position) or sensors (e.g., accelerometers [53]) in order for their computational representation to match. If modules do not have enough input or processing capabilities, this can be externalized (e.g., projector-based tracking [57]) although this prevents mobility.

Dimensionality ← *Shape.* The dimensionality depends on the shape of the modules. E.g., cylinders will allow for 2D UIs [57, 93] and quasi-spheres or cube allow for 3D UIs [81, 89].

Usage consumption ← *Power storage.* The higher the power storage, the higher the autonomy. However, a way of expanding usage time is to limit the energy consumption *between* reconfigurations, with modules able to retain their shape and interactive capabilities with little to no power (multistability) [91]. This allow modules to save and provide enough power for the reconfiguration. Although all surveyed 2D swarm-based implementations are battery-powered and can keep their shape without consuming energy, 3D drone-based swarm UIs [10] cannot as they need power to keep levitating. As drones are currently the only way of achieving 3D in swarm systems, only 2D swarm implementations are currently multistable. Mains-operated systems do not have to deal with autonomy issues, but are not transportable as they need to be constantly plugged-in. This could especially be an issue with systems targeting handheld mobile scenarii (e.g., [72, 73]).

7 DISCUSSION

We discuss the opportunities and limitations of our structured cross-disciplinary space for modular shape-changing UIs. To do this, we systematically assess it through the descriptive, evaluative and generative powers from [6], and reports on its limitations.

7.1 Descriptive, evaluative and generative powers

Our contribution allows to describe, compare and generate new designs. These benefits are important to the HCI community [6]. To illustrate these three powers, we take the example of two similar interactive chain-based implementations: chainFORM [72] and lineFORM [73]. First, we demonstrate the descriptive and evaluative powers of our structured space in Table 3. If we had to do the description and comparison of the macro-properties without our structured space, we would have to use at least 9 different papers: [47] (shape-changing ability), [86] (control over shape change), [87] (interactivity+reversibility), [9, 38] (combination between states), [2] (usage consumption + safety + resolution), [40, 41](detailed safety), [91] (reconfiguration volume + smoothness), [27] (hierarchy), and [103] (dimensionality). The table shows that the structured cross-disciplinary space allows to

	Properties			lineFORM [73]	chainFORM [72]
Micro-properties			I/O	**none**	**touch sensors + LEDs**
			Processing	**limited (externalized Arduino Mega)**	**limited (externalized Teensy 3.2)**
			Power Storage	mains-operated	mains-operated
			Size	**32x70x40mm**	**25x30x12mm**
			Shape	**rect. parallellepiped w/ bracket**	**0.8kg.cm**
			Speed	0.103 sec/60 deg	0.10 sec/60 deg
			Density	**lower (bigger modules)**	**higher (smaller modules)**
			Structure	chained	chained
			Strength	**8.3kg.cm**	**0.8kg.cm**
Macro-properties	Coupling		Hierarchy	satellites + externalized microcontroller	satellites + externalized microcontroller
			Smoothness	**seamless, fabric cover**	**not seamless, gaps between modules**
			Resolution	**lower (bigger modules)**	**higher (smaller modules + 8 LEDs)**
	Combination between states			Yes	Yes
	Shape-change ability		Amplitude	**lower (17 modules)**	**higher (max. 32 modules)**
			Zero-crossing	**0-8 (17 modules)**	**0-16 (max. 32 modules)**
			Curvature	**higher (0–232°between two modules)**	**lower (0–119.5°between two modules)**
			Speed	0.103 sec/60°	0.10 sec/60°
			Modulartiy	No	No
			Porosity	No	No
			Stretchability	No	No
			Granularity	**max. 0.04 cp/cm2**	**max. 0.13 cp/cm2**
			Strength	not mentioned	not mentioned
		Size	**Length**	**Yes (max. 186cm)**	**Yes (max not mentioned)**
			Area	Yes (max not mentioned)	Yes (max not mentioned)
			Volume	Yes (max not mentioned)	Yes (max not mentioned)
	Reversibility			Yes	Yes
	Dimensionality			**3D**	**2D***
	Volume for shape-change			not mentioned	not mentioned
	Usage consumption			**not mentioned**	**83mA.h per module**
	Control over shape-change			negotiated	negotiated
	Interactivity			**shape-change**	**shape-change + touch + LEDs**
	Safety		Severity	not mentioned	not mentioned
			Avoidance	not mentioned	not mentioned
			Redundancy	not mentioned	not mentioned

Table 3. Description and comparison of two chain-based interactive modular shape-changing implementations: lineFORM [73] and chainFORM [72]. Items in bold highlight the differences between the two systems.
**3D possible but limited, only through direct control: the modules reconfigure on a 2D plan, the user needs to detach the modules, re-arrange them with a plastic joint between them to locate them in two different 2D planes.*

describe both systems in details. The items in bold in the table also show the difference between both systems. The structured space allows a fine comparison between these systems, despite the difficulty as they are very similar.

Second, we demonstrate the *generative power* of our contribution by modifying a property of chainFORM and observe the resulting direct and indirect impact on macro- and micro-properties. At first glance, it would seem fairly easy to implement *Modularity* (i.e. the ability to detach part of the chain while mainting each subchain

functional) with chainFORM, as the modules can readily be detached through direct shape-deformation. However, our structured space shows that:
1) A change in the *modularity* of the UI would impact the *hierarchy* of the UI (Figure 3).
Both subchains need at least a standalone to operate, which is not the case in the current implementation.
2) A change in hierarchy of the UI would impact *power storage* of the modules (Figure 3).
3) A change in power storage of the modules would impact the *strength* of the modules (Figure 4).
4-a) A change in the strength of the modules would impact *shape-changing abilities* of the UI (Figure 2).
4-b) A change in the strength of the modules would impact the *speed* of the modules (Figure 4), as each subchain will have less modules for the same power, hence can be faster.
5) A change in the speed of the modules would impact the speed of the UI, i.e., its *shape-change ability* (Figure 2).

As a result, our contribution helps HCI researchers to generate new designs by exploring more easily modifications of existing ones. To this aim, we provide an overview of the direct and indirect impacts of potential design choices at the micro- and macro-scale. Our contribution also helps HCI researchers to generate new designs from scratch by considering all properties one after the other, or further improve existing ones, through an understanding of the design properties for modular shape-changing UIs and how they may impact usability.

7.2 Limitations

Reconfiguration algorithms. We do not discuss in this paper the challenges of user-centered reconfiguration algorithms [91, 122]. Although this is an interesting topic, we leave it for future work, and propose a structure that can easily accommodate such an extension in the future. Indeed, the current list of macro-scale properties of the entire interface can easily be extended, further detailing digital properties (Figure 3). Future work should address them when the hardware barriers are broken [2].

Interactivity. The *Interactivity* property (section 4.9) is currently sufficient to compare existing systems, as currently few of them are interactive. Future work can easily further detail the interaction modalities, for instance leveraging a definition of interaction modality [75].

Safety. The *Safety* (section 4.10) is currently seldom addressed in the literature. In this paper, we used standards from the robotics and HRI communities to describe this property, but those are not specific to modular implementations. We expect that new safety concerns will arise as modular shape-changing interfaces evolve, becoming more robust and allowing researchers to conduct user experiments.

Context of use. A human-centered design approach should start by describing and understanding the context of use before drawing requirements and producing design solutions (bottom-up approach) [39]. However, there is no existing usage of modular shape-changing interfaces [2]: existing implementations are restricted to research prototypes often not robust enough to support user interaction [2]. Therefore, the context of use is unknown (user(s), characteristics of the user(s) or groups of users, goals and tasks of the user(s) and environment(s) of the system [39]). As a consequence, in this paper, we took a *top-down* approach to provide a structured space and inform the early design of modular shape-changing interfaces. Future work should take into consideration the impact of macro- and micro-properties on the contexts of use as they arise, i.e. either existing contexts which could benefit from the implementation of modular UIs (e.g., the same way non-modular shape-changing control devices are studied to replace bulky control interfaces [79, 80]) or new usages uniquely enabled by modular shape-changing UIs.

8 CONCLUSION

We presented a structured property space for the design of modular shape-changing user interfaces. To this end, we conducted a cross-disciplinary, systematic literature review to propose (1) a set of design properties at the

scale of the interface (macro-scale) and at the scale of the modules (micro-scale) and, (2) the impact of these properties on each other.

This work can be *readily used* to describe and compare existing modular shape-changing UIs, allowing practitioners to choose the design which best fit their needs, and generate new design ideas by building upon knowledge from robotics and HCI.

This work can be further *refined* in the future by studying more digital properties (e.g., *Reconfiguration algorithms*). In the mid-term future, the *Interactivity* property could be refined to take into account interaction techniques, as the field recently started to move from solely designing hardware and software, to designing interaction techniques (e.g., [50, 52, 105]). We will also further refine our work in the future by putting it to use for the design of new modular shape-changing UIs and further strengthen its confidence. In the long-term, we believe that researchers will conduct more user experiments as the field matures, the technology advances and the prototypes become more robust. User experiments will allow us to evaluate the impact of our properties on user experience, as well as their impact on each other. For example, what strength do users apply on a given modular interface? How do they grasp it? What is the impact on our macro- and micro-scale properties on acceptability? The structured, cross-disciplinary space of shape-changing modular user interfaces aims at helping the HCI community to achieve these goals.

ACKNOWLEDGMENTS

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program *Investissement d'avenir*.

REFERENCES

[1] Reem J. Alattas, Sarosh Patel, and Tarek M. Sobh. 2019. Evolutionary Modular Robotics: Survey and Analysis. 95, 3 (2019), 815–828. https://doi.org/10.1007/s10846-018-0902-9

[2] Jason Alexander, Anne Roudaut, Jürgen Steimle, Kasper Hornbæk, Miguel Bruns Alonso, Sean Follmer, and Timothy Merritt. 2018. Grand Challenges in Shape-Changing Interface Research. In *the CHI 2018 Conference*. New York, New York, USA, 1–14. https://doi.org/10.1145/3173574.3173873

[3] Apple. 2020. *iPad 10.2-inch - Technical Specifications*. https://www.apple.com/ipad-10.2/specs/ Library Catalog: www.apple.com.

[4] Nora Ayanian, Paul J. White, Adam Halasz, Mark Yim, and Vijay Kumar. 2008. Stochastic Control for Self-Assembly of XBots. In *Volume 2: 32nd Mechanisms and Robotics Conference, Parts A and B* (Brooklyn, New York, USA, 2008-01-01). ASMEDC, 1169–1176. https://doi.org/10.1115/DETC2008-49535

[5] Christian Bastien and Dominique L Scapin. 1993. Ergonomic criteria for the evaluation of human-computer interfaces. (1993), 83.

[6] Michel Beaudouin-Lafon. 2004. Designing Interaction, Not Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (Gallipoli, Italy) *(AVI '04)*. Association for Computing Machinery, New York, NY, USA, 15–22. https://doi.org/10.1145/989863.989865

[7] Julien Bourgeois. 2018. Programmable matter: forming objects with modular robots. In *14th International Symposium on Distributed Autonomous Robotic Systems (2018)*. Boulder, CO, United States.

[8] J. Bourgeois and S. C. Goldstein. 2015. Distributed Intelligent MEMS: Progresses and Perspectives. *IEEE Systems Journal* 9, 3 (Sep. 2015), 1057–1068. https://doi.org/10.1109/JSYST.2013.2281124

[9] Julien Bourgeois, Benoît Piranda, André Naz, Nicolas Boillot, Hakim Mabed, Dominique Dhoutaut, Thadeu Tucci, and Hicham Lakhlef. 2016. Programmable matter as a cyber-physical conjugation. *SMC* (2016). https://dblp.org/rec/conf/smc/BourgeoisPNBMDT16

[10] Sean Braley, Calvin Rubens, Timothy R. Merritt, and Roel Vertegaal. 2018. GridDrones: A Self-Levitating Physical Voxel Lattice for 3D Surface Deformations. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18* (Montreal QC, Canada, 2018). ACM Press, 1–4. https://doi.org/10.1145/3170427.3186477

[11] Manuele Brambilla, Arne Brutschy, Marco Dorigo, and Mauro Birattari. 2014. Property-Driven Design for Robot Swarms: A Design Method Based on Prescriptive Modeling and Model Checking. 9, 4 (2014), 17:1–17:28. https://doi.org/10.1145/2700318

[12] Byoung Kwon An. 2008. Em-cube: cube-shaped, self-reconfigurable robots sliding on structure surfaces. In *2008 IEEE International Conference on Robotics and Automation* (Pasadena, CA, USA, 2008-05). IEEE, 3149–3155. https://doi.org/10.1109/ROBOT.2008.4543690

[13] C J Cascio and K Sathian. 2001. Temporal cues contribute to tactile perception of roughness. *The Journal of neuroscience : the official journal of the Society for Neuroscience* 21, 14 (July 2001), 5289–5296. http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=

pubmed&id=11438604&retmode=ref&cmd=prlinks

[14] Marcelo Coelho and Jamie Zigelbaum. 2011. Shape-changing interfaces. 15, 2 (2011), 161–173. https://doi.org/10.1007/s00779-010-0311-y

[15] Céline Coutrix and Laurence Nigay. 2011. OP: A Novel Programming Model for Integrated Design and Prototyping of Mixed Objects. In *Human-Computer Interaction – INTERACT 2011*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 54–72.

[16] J. Davey, N. Kwok, and M. Yim. 2012. Emulating self-reconfigurable robots - design of the SMORES system. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 4464–4469. https://doi.org/10.1109/IROS.2012.6385845

[17] Felecia Davis, Asta Roseway, Erin Carroll, and Mary Czerwinski. 2013. Actuating mood: design of the textile mirror. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction - TEI '13* (Barcelona, Spain, 2013). ACM Press, 99. https://doi.org/10.1145/2460625.2460640

[18] Marco Dorigo, Dario Floreano, Luca Maria Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy, Daniel Burnier, Alexandre Campo, Anders Lyhne Christensen, Antal Decugniere, Gianni Di Caro, Frederick Ducatelle, Eliseo Ferrante, Alexander Forster, Javier Martinez Gonzales, Jerome Guzzi, Valentin Longchamp, Stephane Magnenat, Nithin Mathews, Marco Montes de Oca, Rehan O'Grady, Carlo Pinciroli, Giovanni Pini, Philippe Retornaz, James Roberts, Valerio Sperati, Timothy Stirling, Alessandro Stranieri, Thomas Stutzle, Vito Trianni, Elio Tuci, Ali Emre Turgut, and Florian Vaussard. 2013. Swarmanoid: A Novel Concept for the Study of Heterogeneous Robotic Swarms. 20, 4 (2013), 60–71. https://doi.org/10.1109/MRA.2013.2252996

[19] Frederick Ducatelle, Gianni A. Di Caro, and Luca M. Gambardella. 2010. Cooperative self-organization in a heterogeneous swarm robotic system. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation* (New York, NY, USA, 2010-07-07) *(GECCO '10)*. Association for Computing Machinery, 87–94. https://doi.org/10.1145/1830483.1830501

[20] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. 1993. A taxonomy for swarm robots. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, Vol. 1. IEEE, 441–447.

[21] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. 1995. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 442–449. https://doi.org/10.1145/223904.223964

[22] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation. In *UIST '13: Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, New York, New York, USA, 417–426. https://doi.org/10.1145/2501988.2502032

[23] Toshio Fukuda and Yoshio Kawauchi. 1990. Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, USA, May 13-18, 1990*. 662–667. https://doi.org/10.1109/ROBOT.1990.126059

[24] Marwan Gharbia, Alice Chang-Richards, Yuqian Lu, Ray Y. Zhong, and Heng Li. 2020. Robotic technologies for on-site building construction: A systematic review. *Journal of Building Engineering* 32 (Nov. 2020), 101584. https://doi.org/10.1016/j.jobe.2020.101584

[25] Kyle Gilpin, Ara Knaian, and Daniela Rus. 2010. Robot pebbles: One centimeter modules for programmable matter through self-disassembly. In *2010 IEEE International Conference on Robotics and Automation* (Anchorage, AK, 2010-05). IEEE, 2485–2492. https://doi.org/10.1109/ROBOT.2010.5509817

[26] Kyle Gilpin, Keith Kotay, Daniela Rus, and Iuliu Vasilescu. 2008. Miche: Modular Shape Formation by Self-Disassembly. 27, 3 (2008), 345–372. https://doi.org/10.1177/0278364907085557

[27] Alix Goguey, Cameron Steer, Andrés Lucero, Laurence Nigay, Deepak Ranjan Sahoo, Céline Coutrix, Anne Roudaut, Sriram Subramanian, Yutaka Tokuda, Timothy Neate, and et al. 2019. PickCells: A Physically Reconfigurable Cell-Composed Touchscreen. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 273, 14 pages. https://doi.org/10.1145/3290605.3300503

[28] Antonio Gomes, Calvin Rubens, Sean Braley, and Roel Vertegaal. 2016. BitDrones: Towards Using 3D Nanocopter Displays As Interactive Self-Levitating Programmable Matter. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 770–780. https://doi.org/10.1145/2858036.2858519

[29] Chase Greenhagen, Timothy Krentz, Janelle Wigal, and Sami Khorbotly. 2016. A real-life robotic application of the particle swarm optimization algorithm. In *2016 Swarm/Human Blended Intelligence Workshop (SHBI)* (2016-10). 1–5. https://doi.org/10.1109/SHBI.2016.7780281

[30] Roderich Groß and Marco Dorigo. 2008. Self-Assembly at the Macroscopic Scale. 96 (2008), 1490–1508. https://doi.org/10.1109/JPROC.2008.927352

[31] Grünbaum. 1994. Uniform tilings of 3-space. *Geombinatorics* 4 (1994). Issue 2.

[32] Bahar Haghighat, Emmanuel Droz, and Alcherio Martinoli. 2015. Lily: A miniature floating robotic platform for programmable stochastic self-assembly. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015-05). 1941–1948. https://doi.org/10.1109/ICRA.2015.7139452 ISSN: 1050-4729.

[33] Takefumi Hiraki, Shogo Fukushima, and Takeshi Naemura. 2016. Phygital field: an integrated field with a swarm of physical robots and digital images. In *SIGGRAPH ASIA 2016 Emerging Technologies* (New York, NY, USA, 2016-11-28) *(SA '16)*. Association for Computing Machinery, 1–2. https://doi.org/10.1145/2988240.2988242

[34] David Holman and Roel Vertegaal. 2008. Organic user interfaces: designing computers in any way, shape, or form. 51, 6 (2008), 48. https://doi.org/10.1145/1349026.1349037

[35] Hongxing Wei, Yingpeng Cai, Haiyuan Li, Dezhong Li, and Tianmiao Wang. 2010. Sambot: A self-assembly modular robot for swarm robot. In *2010 IEEE International Conference on Robotics and Automation* (Anchorage, AK, 2010-05). IEEE, 66–71. https://doi.org/10.1109/ROBOT.2010.5509214

[36] J.R. Hook and H.E. Hall. 1991. *Solid State Physics, 2nd Edition*. Wiley.

[37] K. Hosokawa, T. Tsujimori, T. Fujii, H. Kaetsu, H. Asama, Y. Kuroda, and I. Endo. 1998. Self-organizing collective robots with morphogenesis in a vertical plane. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)* (1998-05), Vol. 4. 2858–2863 vol.4. https://doi.org/10.1109/ROBOT.1998.680616 ISSN: 1050-4729.

[38] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical Atoms: Beyond Tangible Bits, toward Transformable Materials. *Interactions* 19, 1 (Jan. 2012), 38–51. https://doi.org/10.1145/2065327.2065337

[39] ISO9241-210:2019 2019. *Ergonomics of Human-System Interaction – Part 210: Human-Centered Design for Interactive Systems*. Standard. International Organization for Standardization, Geneva, CH.

[40] ISO/DIS 10218-1 2018. *Robotics — Safety requirements for robot systems in an industrial environment — Part 1: Robots*. Standard. International Organization for Standardization, Geneva, CH.

[41] ISO/TS 15066:2016. 2016. *Robots and robotic devices — Collaborative robots*. Standard. International Organization for Standardization, Geneva, CH.

[42] Yunyao Jiang and Yaning Li. 2017. 3D Printed Chiral Cellular Solids with Amplified Auxetic Effects Due to Elevated Internal Rotation. *Advanced Engineering Materials* 19, 2 (2017), 1600609. https://doi.org/10.1002/adem.201600609

[43] Akiya Kamimura, Eiichi Yoshida, Satoshi Murata, Haruhisa Kurokawa, Kohji Tomita, and Shigeru Kokaji. 2002. A Self-Reconfigurable Modular Robot (MTRAN) — Hardware and Motion Planning Software —. In *Distributed Autonomous Robotic Systems 5* (Tokyo, 2002), Hajime Asama, Tamio Arai, Toshio Fukuda, and Tsutomu Hasegawa (Eds.). Springer Japan, 17–26. https://doi.org/10.1007/978-4-431-65941-9_3

[44] M. E. Karagozler, J. D. Campbell, G. K. Fedder, S. C. Goldstein, M. P. Weller, and B. W. Yoon. 2007. Electrostatic latching for inter-module adhesion, power transfer, and communication in modular robots. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007-10). 2779–2786. https://doi.org/10.1109/IROS.2007.4399492 ISSN: 2153-0866.

[45] Mustafa E Karagozler, Brian Kirby, Wei Jie Lee, Eugene Marinelli, Tze Chang Ng, Michael P Weller, and Seth C Goldstein. 2006. Ultralight Modular Robotic Building Blocks for the Rapid Deployment of Planetary Outposts. (2006), 15.

[46] Hamidreza Karbasi, Jan Paul Huissoon, and Amir Khajepour. 2004. Uni-drive modular robots: theory, design, and experiments. 39, 2 (2004), 183–200. https://doi.org/10.1016/S0094-114X(03)00113-7

[47] Hyunyoung Kim, Céline Coutrix, and Anne Roudaut. 2018. Morphees+: Studying Everyday Reconfigurable Objects for the Design and Taxonomy of Reconfigurable UIs. In *CHI '18*. ACM Press, New York, New York, USA, 1–14. https://doi.org/10.1145/3173574.3174193

[48] Hyunyoung Kim, Céline Coutrix, and Anne Roudaut. 2019. KnobSlider: Design of a Shape-Changing Parameter Control UI and Study of User Preferences on Its Speed and Tangibility. 6 (2019), 79. https://doi.org/10.3389/frobt.2019.00079

[49] Hyunyoung Kim, Patrícia Deud Guimarães, Céline Coutrix, and Anne Roudaut. 2019. ExpanDial: Designing a Shape-Changing Dial. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (San Diego, CA, USA) *(DIS '19)*. Association for Computing Machinery, New York, NY, USA, 949–961. https://doi.org/10.1145/3322276.3322283

[50] Lawrence H Kim, Daniel S Drew, Veronika Domova, and Sean Follmer. 2020. User-defined Swarm Robot Control. (2020), 13.

[51] Lawrence H. Kim and Sean Follmer. 2017. UbiSwarm: Ubiquitous Robotic Interfaces and Investigation of Abstract Motion as a Display. 1, 3 (2017), 1–20. https://doi.org/10.1145/3130931

[52] Lawrence H. Kim and Sean Follmer. 2019. SwarmHaptics: Haptic Display with Swarm Robots. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19* (Glasgow, Scotland Uk, 2019). ACM Press, 1–13. https://doi.org/10.1145/3290605.3300918

[53] Brian T. Kirby, Michael Ashley-Rollman, and Seth Copen Goldstein. 2011. Blinky blocks: a physical ensemble programming platform. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11* (Vancouver, BC, Canada, 2011). ACM Press, 1111. https://doi.org/10.1145/1979742.1979712

[54] Kurt Konolige, Charlie Ortiz, Regis Vincent, Benoit Morisset, Andrew Agno, Michael Eriksen, Dieter Fox, Benson Limketkai, Jonathan Ko, Benjamin Stewart, and Dirk Schulz. 2004. Centibots: Very Large Scale Distributed Robotic Teams. In *Building the Information Society*, Renè Jacquart (Ed.). Vol. 156. Springer US, 761–761. https://doi.org/10.1007/978-1-4020-8157-6_84 Series Title: IFIP International Federation for Information Processing.

[55] K.D. Kotay and D.L. Rus. 1998. Motion synthesis for the self-reconfiguring molecule. In *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)* (Victoria, BC, Canada, 1998), Vol. 2. IEEE, 843–851. https://doi.org/10.1109/IROS.1998.727304

[56] H. Kurokawa, E. Yoshida, K. Tomita, A. Kamimura, S. Murata, and S. Kokaji. 2006. Self-reconfigurable M-TRAN structures and walker generation. 54, 2 (2006), 142–149. https://doi.org/10.1016/j.robot.2005.09.023

[57] Mathieu Le Goc, Lawrence H Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building Blocks for Swarm User Interfaces. In *UIST'16*. 1–13. https://doi.org/10.1145/2984511.2984547

[58] Dazhai Li, Hualei Fu, and Wei Wang. 2008. Ultrasonic based autonomous docking on plane for mobile robot. In *2008 IEEE International Conference on Automation and Logistics* (2008-09). 1396–1401. https://doi.org/10.1109/ICAL.2008.4636372 ISSN: 2161-816X.

[59] Alessandro Liberati, Douglas G. Altman, Jennifer Tetzlaff, Cynthia Mulrow, Peter C. Gøtzsche, John P. A. Ioannidis, Mike Clarke, P. J. Devereaux, Jos Kleijnen, and David Moher. 2009. The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration. 6, 7 (2009), e1000100. https://doi.org/10.1371/journal.pmed.1000100

[60] Sara Ljungblad and Lars Erik Holmquist. 2005. Designing robot applications for everyday environments. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies* (New York, NY, USA, 2005-10-12) *(sOc-EUSAI '05)*. Association for Computing Machinery, 65–68. https://doi.org/10.1145/1107548.1107571

[61] Pedro Lopes, Patrik Jonell, and Patrick Baudisch. 2015. Affordance++: Allowing Objects to Communicate Dynamic Use. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea, 2015-04-18) *(CHI '15)*. Association for Computing Machinery, 2515–2524. https://doi.org/10.1145/2702123.2702128

[62] Henrik Hautop Lund. 2011. Anybody, anywhere, anytime - Robotics with a social impact through a building block approach. In *Advanced Robotics and its Social Impacts*. 2–7. https://doi.org/10.1109/ARSO.2011.6301970 ISSN: 2162-7576.

[63] Henrik Hautop Lund. 2014. Lessons Learned in Designing User-Configurable Modular Robotics. In *Robot Intelligence Technology and Applications 2: Results from the 2nd International Conference on Robot Intelligence Technology and Applications*, Jong-Hwan Kim, Eric T . Matson, Hyun Myung, Peter Xu, and Fakhri Karray (Eds.). Springer International Publishing, 279–286. https://doi.org/10.1007/978-3-319-05582-4_24

[64] David Merrill, Jeevan J Kalanithi, and Pattie Maes. 2007. Siftables - towards sensor network user interfaces.. In *Tangible and Embedded Interaction*. https://dblp.org/rec/conf/tei/MerrillKM07

[65] G. Michelitsch, J. Williams, M. Osen, B. Jimenez, and S. Rapp. 2004. Haptic Chameleon: A New Concept of Shape-Changing User Interface Controls with Force Feedback. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria) *(CHI EA '04)*. Association for Computing Machinery, New York, NY, USA, 1305–1308. https://doi.org/10.1145/985921.986050

[66] Shuhei Miyashita, Max Lungarella, and Rolf Pfeifer. 2009. Tribolon: Water-Based Self-Assembly Robots. In *Artificial Life Models in Hardware*, Andrew Adamatzky and Maciej Komosinski (Eds.). Springer, 161–184. https://doi.org/10.1007/978-1-84882-530-7_8

[67] Rico Moeckel, Cyril Jaquier, Kevin Drapel, Elmar Dittrich, Andres Upegui, and Auke Jan Ijspeert. 2006. Exploring adaptive locomotion with YaMoR, a novel autonomous modular robot with Bluetooth interface. 33, 4 (2006), 285–290. https://doi.org/10.1108/01439910610667908

[68] Francesco Mondada, Giovanni C. Pettinaro, Andre Guignard, Ivo W. Kwee, Dario Floreano, Jean-Louis Deneubourg, Stefano Nolfi, Luca Maria Gambardella, and Marco Dorigo. 2004. Swarm-Bot: A New Distributed Robotic Concept. 17, 2 (2004), 193–221. https://doi.org/10.1023/B:AURO.0000033972.50769.1c

[69] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji. 1998. A 3-D self-reconfigurable structure. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)* (Leuven, Belgium), Vol. 1. IEEE, 432–439. https://doi.org/10.1109/ROBOT.1998.677012

[70] Satoshi Murata, Eiichi Yoshida, Akiya Kamimura, Kohji Tomita, Haruhisa Kurokawa, and Shigeru Kokaji. 2003. Homogeneous Distributed Mechanical Systems. In *Morpho-functional Machines: The New Species* (Tokyo), Fumio Hara and Rolf Pfeifer (Eds.). Springer Japan, 167–193. https://doi.org/10.1007/978-4-431-67869-4_9

[71] Satoshi Murata, Eiichi Yoshida, Haruhisa Kurokawa, Kohji Tomita, and Shigeru Kokaji. 2001. Concept of self-reconfigurable modular robotic system. 15, 4 (2001), 383–387. https://doi.org/10.1016/S0954-1810(01)00024-3

[72] Ken Nakagaki, Artem Dementyev, Sean Follmer, Joseph A. Paradiso, and Hiroshi Ishii. 2016. ChainFORM: A Linear Integrated Modular Hardware System for Shape Changing Interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan, 2016-10-16) *(UIST '16)*. Association for Computing Machinery, 87–96. https://doi.org/10.1145/2984511.2984587

[73] K Nakagaki, S Follmer, and H Ishii. 2015. LineFORM: Actuated Curve Interfaces for Display, Interaction, and Constraint. In *Proceedings of the 28th Annual ACM Conference on Human Factors in Computing*. https://doi.org/10.1145/2807442.2807452

[74] Changjoo Nam, Huao Li, Shen Li, Michael Lewis, and Katia Sycara. 2018. Trust of Humans in Supervisory Control of Swarm Robots with Varied Levels of Autonomy. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2018-10). 825–830. https://doi.org/10.1109/SMC.2018.00148 ISSN: 2577-1655.

[75] Laurence Nigay and Joëlle Coutaz. 1995. A Generic Platform for Addressing the Multimodal Challenge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 98–105. https://doi.org/10.1145/223904.223917

[76] Esben Hallundbæk Østergaard, Kristian Kassow, Richard Beck, and Henrik Hautop Lund. 2006. Design of the ATRON lattice-based self-reconfigurable robot. *Autonomous Robots* 21, 2 (01 Sep 2006), 165–183. https://doi.org/10.1007/s10514-006-8546-1

[77] Amit Pamecha and Chih-Jung Chiang. 1996. DESIGN AND IMPLEMENTATION OF METAMORPHIC ROBOTS. (1996), 11.

[78] Pat Pannuto, Yoonmyung Lee, ZhiYoong Foo, Gyouho Kim, David Blaauw, and Prabal Dutta. 2015. Lessons from Five Years of Making Michigan Micro Motes. (2015), 2.

[79] Sylvain Pauchet, Catherine Letondal, Jean-Luc Vinot, Mickaël Causse, Mathieu Cousy, Valentin Becquet, and Guillaume Crouzet. 2018. GazeForm: Dynamic Gaze-adaptive Touch Surface for Eyes-free Interaction in Airliner Cockpits. In *Proceedings of the 2018 on Designing Interactive Systems Conference 2018 - DIS '18* (Hong Kong, China, 2018). ACM Press, 1193–1205. https://doi.org/10.1145/3196709.3196712

[80] Sylvain Pauchet, Jean-Luc Vinot, Catherine Letondal, Alexandre Lemort, Claire Lavenir, Timothée Lecomte, Stéphanie Rey, Valentin Becquet, and Guillaume Crouzet. 2019. Multi-plié: A Linear Foldable and Flattenable Interactive Display to Support Efficiency, Safety and Collaboration. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19* (Glasgow, Scotland Uk, 2019). ACM Press, 1–13. https://doi.org/10.1145/3290605.3300384

[81] Benoit Piranda and Julien Bourgeois. 2016. Geometrical study of a quasi-spherical module for building programmable matter. In *DARS 2016, 13th International Symposium on Distributed Autonomous Robotic Systems*. https://www.researchgate.net/profile/Benoit_Piranda/publication/307930970_Geometrical_study_of_a_quasi-spherical_modules_for_building_programmable_matter/links/582438d308ae7ea5be7233a7.pdf

[82] Benoit Piranda and Julien Bourgeois. 2018. Designing a quasi-spherical module for a huge modular robot to create programmable matter. *Autonomous Robot Journal, Special Issue: Distributed Robotics: From Fundamentals to Applications* 42, 8 (2018), 1619–1633. https://doi.org/10.1007/s10514-018-9710-0

[83] Benoit Piranda and Julien Bourgeois. 2020. Datom: A Deformable modular robot for building self-reconfigurable programmable matter. (2020). arXiv:2005.03402 http://arxiv.org/abs/2005.03402

[84] Ivan Poupyrev, Tatsushi Nashida, and Makoto Okabe. 2007. Actuation and Tangible User Interfaces: the Vaucanson Duck, Robots, and Shape Displays. (2007), 8.

[85] Jorge Peña Queralta, Li Qingqing, Tuan Nguyen Gia, Hong-Linh Truong, and Tomi Westerlund. 2020. End-to-End Design for Self-Reconfigurable Heterogeneous Robotic Swarms. In *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (2020-05). 281–287. https://doi.org/10.1109/DCOSS49796.2020.00052 ISSN: 2325-2944.

[86] Majken Kirkegård Rasmussen, Timothy Merritt, Miguel Bruns Alonso, and Marianne Graves Petersen. 2016. Balancing User and System Control in Shape-Changing Interfaces: A Designerly Exploration. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (Eindhoven, Netherlands) *(TEI '16)*. Association for Computing Machinery, New York, NY, USA, 202–210. https://doi.org/10.1145/2839462.2839499

[87] Majken K. Rasmussen, Esben W. Pedersen, Marianne G. Petersen, and Kasper Hornbæk. 2012. Shape-changing interfaces: a review of the design space and open research questions. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12* (Austin, Texas, USA, 2012). ACM Press, 735. https://doi.org/10.1145/2207676.2207781

[88] Simon Robinson, Céline Coutrix, Jennifer Pearson, Juan Rosso, Matheus Fernandes Torquato, Laurence Nigay, and Matt Jones. 2016. Emergeables: Deformable Displays for Continuous Eyes-Free Mobile Interaction. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Swansea University, ACM, New York, New York, USA, 3793–3805. https://doi.org/10.1145/2858036.2858097

[89] John W. Romanishin, Kyle Gilpin, Sebastian Claici, and Daniela Rus. 2015. 3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA, USA, 2015-05). IEEE, 1925–1932. https://doi.org/10.1109/ICRA.2015.7139450

[90] J. W. Romanishin, K. Gilpin, and D. Rus. 2013. M-blocks: Momentum-driven, magnetic modular robots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 4288–4295. https://doi.org/10.1109/IROS.2013.6696971

[91] A Roudaut, D Krusteva, M McCoy, A Karnik, K Ramani, and S Subramanian. 2016. Cubimorph: Designing modular interactive devices. In *2016 IEEE International Conference on Robotics and Automation (ICRA*. IEEE, 3339–3345. https://doi.org/10.1109/ICRA.2016.7487508

[92] Anne Roudaut, Henning Pohl, and Patrick Baudisch. 2011. Touch input on curved surfaces. In *CHI '11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/1978942.1979094

[93] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. 2012. Kilobot: A low cost scalable robot system for collective behaviors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 3293–3298.

[94] M. Rubenstein, K. Payne, P. Will, and Wei-Min Shen. 2004. Docking among independent and autonomous CONRO self-reconfigurable robots. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004* (New Orleans, LA, USA, 2004). IEEE, 2877–2882 Vol.3. https://doi.org/10.1109/ROBOT.2004.1307497

[95] Daniela Rus and Marsette Vona. 2001. Crystalline Robots: Self-Reconfiguration with Compressible Unit Modules. 10, 1 (2001), 107–124. https://doi.org/10.1023/A:1026504804984

[96] G. G. Ryland and H. H. Cheng. 2010. Design of iMobot, an intelligent reconfigurable mobile robot with novel locomotion. In *2010 IEEE International Conference on Robotics and Automation*. 60–65.

[97] David Saldana, Bruno Gabrich, Guanrui Li, Mark Yim, and Vijay Kumar. 2018. ModQuad: The Flying Modular Structure that Self-Assembles in Midair. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD, 2018-05). IEEE, 691–698. https://doi.org/10.1109/ICRA.2018.8461014

[98] Behnam Salemi, Mark Moll, and Wei-min Shen. 2006. SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing, China, 2006-10). IEEE, 3636–3641. https://doi.org/10.1109/IROS.2006.281719

[99] Orit Shaer and Eva Hornecker. 2010. *Tangible User Interfaces: Past, Present and Future Directions*. Vol. 3. Now Publishers Inc. https://doi.org/10.1561/1100000026

[100] A. Sprowitz, R. Moeckel, M. Vespignani, S. Bonardi, and A.J. Ijspeert. 2014. Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot. *Robotics and Autonomous Systems* 62, 7 (2014), 1016 – 1033. https://doi.org/10.1016/j.robot.2013.08.011 Reconfigurable Modular Robotics.

[101] David St-Onge, Ulysse Côté-Allard, Kyrre Glette, Benoit Gosselin, and Giovanni Beltrame. 2019. Engaging with Robotic Swarms: Commands from Expressive Motion. 8, 2 (2019), 11:1–11:26. https://doi.org/10.1145/3323213

[102] Kasper Stoy and Haruhisa Kurokawa. 2011. Current topics in classic self-reconfigurable robot research. In *Proceedings of the IROS Workshop on Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems*.

[103] Miriam Sturdee and Jason Alexander. 2018. Analysis and Classification of Shape-Changing Interfaces for Design and Application-based Research. 51, 1 (2018), 1–32. https://doi.org/10.1145/3143559

[104] Miriam Sturdee, Aluna Everitt, Joseph Lindley, Paul Coulton, and Jason Alexander. 2019. Visual Methods for the Design of Shape-Changing Interfaces. In *Human-Computer Interaction – INTERACT 2019* (Cham, 2019) *(Lecture Notes in Computer Science)*, David Lamas, Fernando Loizides, Lennart Nacke, Helen Petrie, Marco Winckler, and Panayiotis Zaphiris (Eds.). Springer International Publishing, 337–358. https://doi.org/10.1007/978-3-030-29387-1_19

[105] Ryo Suzuki, Hooman Hedayati, Clement Zheng, James L. Bohn, Daniel Szafir, Ellen Yi-Luen Do, Mark D. Gross, and Daniel Leithinger. 2020. RoomShift: Room-scale Dynamic Haptics for VR with Furniture-moving Swarm Robots. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA, 2020-04-21) *(CHI '20)*. Association for Computing Machinery, 1–11. https://doi.org/10.1145/3313831.3376523

[106] Ryo Suzuki, Junichi Yamaoka, Daniel Leithinger, Tom Yeh, Mark D. Gross, Yoshihiro Kawahara, and Yasuaki Kakehi. 2018. Dynablock: Dynamic 3D Printing for Instant and Reconstructable Shape Formation. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. ACM, New York, NY, USA, 99–111. https://doi.org/10.1145/3242587.3242659

[107] Ryo Suzuki, Clement Zheng, Yasuaki Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D. Gross, and Daniel Leithinger. 2019. ShapeBots: Shape-changing Swarm Robots. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 493–505. https://doi.org/10.1145/3332165.3347911

[108] Ning Tan, Abdullah Aamir Hayat, Mohan Rajesh Elara, and Kristin L. Wood. 2020. A Framework for Taxonomy and Evaluation of Self-Reconfigurable Robotic Systems. 8 (2020), 13969–13986. https://doi.org/10.1109/ACCESS.2020.2965327

[109] Pierre Thalamy, Benoit Piranda, and Julien Bourgeois. 2019. Distributed Self-Reconfiguration using a Deterministic Autonomous Scaffolding Structure. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. Montreal QC, Canada, 140–148.

[110] Pierre Thalamy, Benoît Piranda, and Julien Bourgeois. 2019. A survey of autonomous self-reconfiguration methods for robot-based programmable matter. 120 (2019), 103242. https://doi.org/10.1016/j.robot.2019.07.012

[111] Pierre Thalamy, Benoit Piranda, Frederic Lassabe, and Julien Bourgeois. 2019. Scaffold-Based Asynchronous Distributed Self-Reconfiguration By Continuous Module Flow. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau, China, 2019-11). IEEE, 4840–4846. https://doi.org/10.1109/IROS40897.2019.8967775

[112] John Tiab and Kasper Hornbæk. 2016. Understanding Affordance, System State, and Feedback in Shape-Changing Buttons. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2752–2763. https://doi.org/10.1145/2858036.2858350

[113] Kohji Tomita, Satoshi Murata, Eiichi Yoshida, Haruhisa Kurokawa, and Shigeru Kokaji. 1996. *Reconfiguration Method for a Distributed Mechanical System*. 17–25. https://doi.org/10.1007/978-4-431-66942-5_3

[114] E. Tsykunov, R. Agishev, R. Ibrahimov, L. Labazanova, A. Tleugazy, and D. Tsetserukou. 2019. SwarmTouch: Guiding a Swarm of Micro-Quadrotors With Impedance Control Using a Wearable Tactile Interface. 12, 3 (2019), 363–374. https://doi.org/10.1109/TOH.2019.2927338 Conference Name: IEEE Transactions on Haptics.

[115] Luis Vega, Devin Hughes, Camilo Buscaron, Dr Eric M Schwartz, and Dr A Antonio Arroyo. 2008. MILyBots: Design and Development of Swarm-Robots. (2008), 9.

[116] Santiago Villarreal-Narvaez, Jean Vanderdonckt, Radu-Daniel Vatavu, and Jacob O. Wobbrock. 2020. A Systematic Review of Gesture Elicitation Studies: What Can We Learn from 216 Studies?. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven Netherlands, 2020-07-03). ACM, 855–872. https://doi.org/10.1145/3357236.3395511

[117] Xiaofeng Wang, Minglu Zhang, and Weimin Ge. 2016. A Novel Docking Mechanism Design and Dynamic Performance Analysis of Self-reconfigurable Modular Robot. In *Advances in Reconfigurable Mechanisms and Robots II* (Cham) *(Mechanisms and Machine Science)*, Xilun Ding, Xianwen Kong, and Jian S. Dai (Eds.). Springer International Publishing, 681–692. https://doi.org/10.1007/978-3-319-23327-7_58

[118] Michael Philetus Weller, Mark D. Gross, and Seth Copen Goldstein. 2011. Hyperform specification: designing and interacting with self-reconfiguring materials. 15, 2 (2011), 133–149. https://doi.org/10.1007/s00779-010-0315-7

[119] Paul White, Victor Zykov, Josh Bongard, and Hod Lipson. 2005. Three Dimensional Stochastic Reconfiguration of Modular Robots. In *Robotics: Science and Systems I* (2005-06-08). Robotics: Science and Systems Foundation. https://doi.org/10.15607/RSS.2005.I.022

[120] Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. PneUI: pneumatically actuated soft composite materials for shape changing interfaces. In *UIST '13: Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, New York, New York, USA, 13–22. https://doi.org/10.1145/2501988.2502037

[121] M. Yim, D. G. Duff, and K. D. Roufas. 2000. PolyBot: a modular reconfigurable robot. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, Vol. 1. 514–520 vol.1. https://doi.org/10.1109/ROBOT.2000.844106

[122] M. Yim, W. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian. 2007. Modular Self-Reconfigurable Robot Systems [Grand Challenges of Robotics]. *IEEE Robotics Automation Magazine* 14, 1 (March 2007), 43–52. https://doi.org/10.1109/MRA.2007.339623

[123] Mark Yim, Paul White, Michael Park, and Jimmy Sastra. 2009. Modular Self-Reconfigurable Robots. In *Encyclopedia of Complexity and Systems Science*, Robert A. Meyers (Ed.). Springer, 5618–5631. https://doi.org/10.1007/978-0-387-30440-3_334

[124] Mark H. Yim, John O. Lamping, and Eric W. Mao. 2001. Touchable user interface using self movable robotic modules. http://www.freepatentsonline.com/6243622.html

[125] H.X. Zhang, J. Gonzalez-Gomez, S.Y. Chen, and J.W. Zhang. 2009. Embedded intelligent capability of a modular robotic system. In *2008 IEEE International Conference on Robotics and Biomimetics* (Bangkok, 2009-02). IEEE, 2061–2066. https://doi.org/10.1109/ROBIO.2009.4913319

[126] Yiwei Zhao, Lawrence H. Kim, Ye Wang, Mathieu Le Goc, and Sean Follmer. 2017. Robotic Assembly of Haptic Proxy Objects for Tangible Interaction and Virtual Reality. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. Association for Computing Machinery, New York, NY, USA, 82–91. https://doi.org/10.1145/3132272.3134143

[127] Victor Zykov, Efstathios Mytilinaios, Mark Desnoyer, and Hod Lipson. 2007. Evolved and Designed Self-Reproducing Modular Robotics. 23, 2 (2007), 308–319. https://doi.org/10.1109/TRO.2007.894685