Automatics: Dynamically Generating Fabrication Tasks to Adapt to Varying Contexts

MATTHEW LAKIER, University of Toronto
MICHELLE ANNETT, MishMashMakers
DANIEL WIGDOR, University of Toronto

When fabricating, it is common to follow a prescribed set of steps in a tutorial or how-to. While popular, such explicit knowledge resources have many inconsistencies and omissions, use static illustrations, and cannot adapt to drop-in makers or a maker's mistakes. To overcome many of these issues, this work presents Automatics, a novel explicit knowledge resource system that dynamically generates fabrication activities for one or more makers based on their current environmental and fabrication context. Automatics assigns tasks to makers based on the past tools and components the maker was working with, enables makers to recover from mistakes through model regeneration, suggests alternative tools if a needed tool is unavailable or in use, and allows multiple makers to drop-in throughout a fabrication activity. Initial usage and feedback from novice makers showed that Automatics increases the number of tasks that can be completed compared to paper instructions, decreases frustration, and improves one's understanding of the global context of assigned tasks during fabrication activities.

CCS Concepts: • **Human-centered computing → Interactive systems and tools**;

Additional Key Words and Phrases: Fabrication, making, assembly, explicit knowledge resources, assembly sequence planning, instruction consumption, error handling, tool substitution, multi-maker fabrication

ACM Reference format:
Matthew Lakier, Michelle Annett, and Daniel Wigdor. 2018. Automatics: Dynamically Generating Fabrication Tasks to Adapt to Varying Contexts. *ACM Trans. Comput.-Hum. Interact.* 25, 4, Article 22 (August 2018), 44 pages.
https://doi.org/10.1145/3185065

1 INTRODUCTION

Whether one belongs to a local makerspace, tinkers in their garage, or prototypes at school, there is a growing ecosystem of makers who want to make, design, prototype, and fabricate. "Maker" is a broad term describing many individuals who may want to learn about new equipment or processes, recreate an artifact they found online, personalize an existing artifact, repair, remix, or recombine multiple artifacts, assemble a piece of furniture, or share their knowledge with others [3, 10, 37, 83,

---

Authors' addresses: M. Lakier and D. Wigdor, DGP Lab, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada; emails: {lakier, daniel}@dgp.toronto.edu; M. Annett, MishMashmakers, Newmarket, Ontario, Canada; email: michelle@mishmashmakers.com.

https://doi.org/10.1145/3185065

84][1]. Regardless of the complexity of the final artifact, fabrication activities often require a maker to (i) draw up plans or designs, (ii) gather materials and tools, (iii) measure or mark out patterns or guides, (iv) cut, drill, shape, or reform existing materials, (v) refine the resulting components, and (vi) assemble the final artifact, all in an interleaved, iterative process.

To ease the complexity inherent in such processes, one typically consults explicit knowledge resources, i.e., online guides such as Instructables, how-to's, or tutorials [10, 75, 82], videos on YouTube [29, 74], or pre-made templates, plans, or guides. Conventionally, such resources provide a maker with the knowledge they require to create an artifact via a prescribed, step-by-step process and can be authored by the original designer, engineer, or maker who created the artifact. These resources often contain static visual 3D information, such as explosion diagrams [73], or pre-recorded, fixed media, such as animations or videos. Much like IKEA or LEGO manuals, many resources do not provide the correct level of detail for a maker [16, 80], have goals that differ from a maker's goals (e.g., makers may wish to modify rather than replicate an existing artifact [53, 72]), and rarely support multiple makers who may wish to assemble or fabricate an artifact together [7, 34].

Despite the efforts of technical writers [31], illustrators [31], and fellow makers [2], recent research has highlighted that such resources often contain errors, with tasks presented in the wrong order or missing, or the wrong tools being specified in the resource [15]. Moreover, makers often do not have the same tools or materials as those specified in a resource [82] or they do not have the knowledge required to use a specified tool or material [15, 82]. Such variability results in fabrication workflows that are non-linear, have uncertainty and uncontrollable variables, and are more context-dependent than the average resource can support [29].

This work seeks to solve many of the challenges that static instruction sets, Instructables, and manuals have by exploring the potential of dynamically-generated explicit knowledge resources, i.e., *Dynamic Manuals*. Dynamic Manuals are dynamic resources that use parametric 3D models and assembly-sequence planning constraints to describe the tasks required to fabricate an artifact. As the creation of instructions is equally as important as their consumption, Automatics seeks to help *designers* who create instructions, such as product designers, technical writers, 3D modelers, engineers, or hobbyists, through a Constraint Specifier application, which enables the information that is needed to create assembly-constraints to be specified in an interactive manner. Once authored, Dynamic Manuals can be used within *Automatics*, a prototype explicit knowledge resource system, to generate and assign fabrication tasks to makers in real-time to handle the variable workflows and uncertain environments that makers encounter while fabricating. Automatics' current implementation is focused on makers who are performing activities that require static explicit knowledge resources to assist with completion: *fabricating an existing artifact* (e.g., following an Instructable, blog post, or video tutorial), *fabricating a variant of an artifact* (e.g., fabricating a design variant of an artifact that is outlined in an Instructable, blog post, or video tutorial), or *assembling a series of components* from a kit (e.g., assembling furniture, LEGO, or a Make: Kit).

Unlike traditional manuals, Automatics supports the dynamism of fabrication-oriented activities, such as cutting, drilling, and joining, which can introduce error, the need for disassembly, or require a maker to find an alternative tool when a needed one is in use or unavailable. This is done using the *Maker Interface*, a dynamic, interactive web-based system that visualizes fabrication and assembly tasks and collects input from a maker about their current environmental context. Automatics assigns and adapts tasks to participating makers using the input from the maker, techniques from assembly-sequence planning, and a heuristic-based Task Assigner module. This architecture

---

[1] Within the context of this work, *maker* refers to those individuals who do not have extensive (or any) experience fabricating objects or assembling components. They are often called casual makers [28].

also enables Automatics to support scenarios where multiple makers may wish to fabricate together, or makers may drop-in or leave a fabrication activity while it is underway.

This work thus enhances and contributes to explicit knowledge resources in three ways. First, a systematic analysis and synthesis of the challenges with explicit knowledge resources is presented. It details a holistic taxonomy of deficiencies within explicit knowledge resources today and informed the design of the prototype system. Second, we present a novel workflow for 'design for making' where the designer of the artifact adds constraints and fabrication-based information to their design, allowing a dynamic instruction system to use this information to guide one or more makers through the making process. There is a specific focus on how the architecture of Automatics uses (i) parametric models to enable an artifact to be regenerated to overcome fabrication errors and support personal design choices, (ii) a constraint satisfaction-based Task Space and Task Assigner to determine all known pathways that can be traversed to fabricate an artifact and assign them to makers in real-time, and (iii) a Maker Interface that accepts contextual information input from a maker and visualizes the past, next, and future tasks that all makers are assigned. Third, we present the results of an initial evaluation of Automatics, which found that participants were able to complete more tasks with less effort and increased motivation using this new method. The study also found that there were several areas that could be fruitful research avenues, such as increasing collaboration between strangers in making tasks and the ability for Automatics to adapt to a particular maker's skillset.

2 RELATED WORK

The most relevant literature to Automatics relates to work on assembly sequence planning (ASP), recommendations for designing static explicit knowledge resources, and interactive assembly systems.

2.1 Assembly Sequence Planning

Many within the manufacturing industry have proposed methods to automatically generate tasks for assembly sequence planning [17, 40]. Homem de Mello and Sanderson applied AND/OR graphs to ASP [27]; instead of providing a method to assemble basic parts, they looked at the final product and the ways it can be disassembled to reduce the number of dead ends reached due to constraints with the product. Del Valle et al. augmented AND/OR graphs with shared resources and task time constraints, enabling for parallel disassembly [17]. Naphade, Storer, and Wu decomposed constraints into smaller "decision dependent constraints" to reduce computation time and allow for the efficient generation of single assembly sequences [50]. Lin and Chang proposed the use of assembly precedence diagrams [40]. The present work is the first to use AND/OR graphs and Del Valle et al.'s notion of parallel disassembly, within the context of fabrication activities. These constructs enable the Dynamic Manuals and Automatics to represent, and quickly traverse, fabrication pathways, whilst treating each indivisible component of the final artifact (i.e., subcomponent) as a resource belonging to a single maker.

Recent work has also focused on ASP for multiple workers. With the Hive project, workers were guided through the construction of a tensegrity module-based pavilion by a "foreman engine" that assigned workers using a lazy heuristic approach [38]. Yoshida et al. used computer vision to find optimal locations for workers to deposit chopsticks to construct a pavilion [91]. Work on disaster response systems and crowdsourcing have also developed methods to assign tasks to workers via task decomposition and resource allocation [13, 18, 42, 43, 57, 63]. The present work is similar, in that Automatics supports multiple makers who may leave or enter the process at any time, however, Automatics differs in that they can also adapt to the mistakes or design variants that often result when a maker manipulates or modifies materials.

2.2 Best Practices for Explicit Knowledge Resources

Most work focused on the design of explicit knowledge resources explores on how to best design static content (e.g., diagrams and text) for consumption during assembly tasks. Early work by Beiger and Glock recommended that the successful assembly of an artifact requires contextual, spatial, and operational information [6]. To better understand how to convey such information, Morrell and Park, for example, compared illustration-only, text-only, and illustration-and-text instructions [49], finding that illustration-and-text instructions decreased assembly errors. Novick and Morse found substantial benefits to step-by-step illustrations over final diagrams of an artifact [51]. Others found benefits to using action diagrams [2], dynamic illustrations [12], and animations [39]. The use of headings [20], high contrast text and images [55], highlighting [35, 55], arrows, frames, and lines [16], and minimizing the number of tasks [45, 55, 59] have also been recommended. The present work followed these recommendations, harnessing animated, annotated, step-by-step illustration-and-text representations within the Maker Interface, and used touch-based controls to allow visualizations of current, past, and future tasks to be easily explored and understood by a maker.

Others have focused on the media most appropriate to present instructions. Zimmerman et al. found that tablets outperformed desktop presentations of instructions [93]. Virtual environments were also found to reduce the errors and decrease assembly times when compared to desktop engineering environments [90] and produced faster assembly times compared to paper instructions [7]. Tang et al. found that head mounted displays decreased assembly errors compared to a monitor [71], whereas Wille et al. found no performance differences but a decrease in physical strain with a head mounted display [88]. When compared to paper, augmented reality has been shown to reduce the number of errors made [4, 41, 70, 87]. Given the prevalence of tablets and laptops, as opposed to VR and AR headsets in makerspaces [48], the Automatics prototype system is currently deployed on a tablet.

2.3 Interactive Assembly Systems

The assembly of components is but one aspect of fabrication; however, many systems have been developed to help with assembly. With the Easy Assemble system [81, 85], for example, a mouse was used to navigate through a prescribed sequence of tasks shown on a desktop PC. The Personal Active Assistant fused CAD assembly software, tagged objects, and a see-through head mounted display to aid in the assembly of manufacturing components [41]. Henderson and Feiner [26] and Ong and Wang [54] removed the need for augmentation by enabling bare-hand, bi-manual part manipulation using computer vision techniques. With TeleAdvisor, Gurivech et al. used a robotic arm, projector, and web camera to enable an expert helper to remotely assist a novice worker [22]. The Duplo-Track system used a Kinect to help assemblers identify the next block to use via visual feedback on a secondary screen [21]. These are but some of the interactive assembly systems that have been created; detailed surveys can be found in [65] and [85, 90]. Although these systems decreased frustration and error during assembly, the focus of the present work is on fabrication, of which assembly is only part of the process. The goal was thus to develop techniques that could support the manipulation of materials, in addition to drop-in makers, tools that become unavailable, and so on.

Interactive workspaces and intelligent tools have been explored to assist with fabrication. Projects such as Drill Sargent [64], SPATA [86], FreeD [94], and the Enchanted Scissors [89] placed focus on the *tools* being used, adding visual and/or haptic feedback to assist with cutting, carving, and so on. With the Smart Makerspace, tools and safety equipment were augmented to enable tool wayfinding and ensure makers adhered to safety protocols [36]. A similar approach was used by

O'Connell with the UTEM equipment management system, which augmented workbenches and tools to collect usage data and "lock out" users for safety reasons [52]. Instead of augmenting a space, Campbell and colleagues augmented a *maker* with sensors to determine if the correct tools were being used for a given task [11]. Antifakos et al. took a hybrid approach, augmenting tools and components so that the current state of an assembled artifact could be recognized [4]. Automatics currently makes use of a manual process to infer the current state of tools (i.e., maker input via the Maker Interface) and the fabricated artifact, but the architecture and implemented techniques could serve as the generation, planning, and assignment modules within the aforementioned systems and tools in the future.

Recently, the WeBuild system by Fraser, Grossman, and Fitzmaurice was proposed as a way to enable multiple co-located users to assemble an artifact using a static, prescribed set of tasks that are created manually [23]. The system made use of smartwatches to show users the current task to perform and provided a dashboard of all assembly tasks on a large display. A task distribution algorithm assigned tasks to makers based on their current speed, how similar tasks were to each other, and if makers should work together or individually, using a task tree that was computed *a priori*. Unlike WeBuild, Automatics focuses on fabrication, where tasks not only require assembly activities, but also the handling and manipulation of raw materials (in addition to the specification of these activities). Automatics also supports the dynamic nature of task assignment (i.e., error recovery and dynamic design variants, alternative tool suggestions, and forfeiting or requesting new tasks due to personal desires or equipment challenges) using techniques from ASP to automatically generate tasks and completion pathways from model constraints, rather than require the entire task representation (and static graphics) to be manually created beforehand, as WeBuild does.

3 DEFICIENCIES IN EXPLICIT KNOWLEDGE RESOURCES

While a recent analysis estimated that 90% of instruction manuals have deficiencies that hamper assembly completion [68], a holistic understanding of the deficiencies inherent in current explicit knowledge resources is missing. Thus, an analysis of the literature related to instructions, how-to's, manuals, procedural sequences, walkthroughs, and so on was conducted using Google Scholar. An open coding process of resulting publications initially classified the deficiencies and a subsequent axial coding activity [69] identified themes crucial to the design of explicit knowledge resources. The deficiencies were then organized into a taxonomy that grounded the focus and design of Automatics (Figure 1).

3.1 Equipment

Many deficiencies are rooted in the equipment that one uses, i.e., the materials, parts, tools, or techniques.

Materials: One of the foremost issues with instruction sets today is that they often omit or mislabel which materials to use for a given step [24, 82]. This can not only lead to confusion and guessing on the part of the maker, but also increases the likelihood that they manipulate the wrong piece of material. Unlike parts like hardware or fasteners, which can be unscrewed or removed if improperly attached, the incorrect manipulation of material could prevent a maker from completing their activity due to a lack of available raw materials, which could thus be costly and time-consuming. If a procedural sequence involves manipulating raw materials that a maker must source themselves, it is common for the materials they have on hand to differ from those specified in the tutorial or instruction set, either in terms of type and properties, e.g., dimension and weight, or for the maker to not have the necessary materials in the first place [62, 82]. When dissimilar materials are available, a maker will often substitute a missing material for one they have on hand,

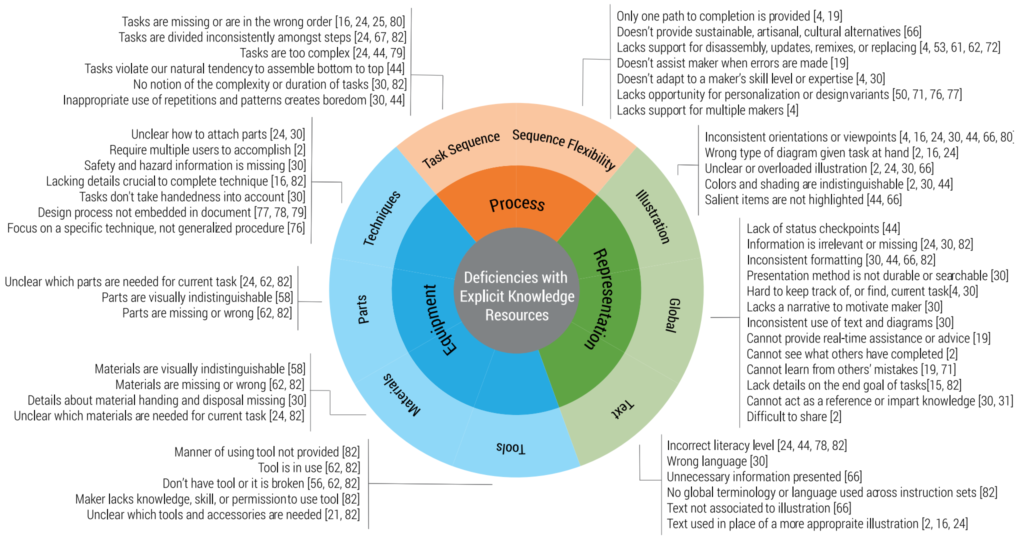


Fig. 1. A taxonomy of the deficiencies found within explicit knowledge resources that emphasizes challenges with the *equipment* to be used, the *processes* that must be undertaken, and the *representation* of the tasks provided.

which could create issues that propagate throughout the rest of their design, for example, cutting one piece smaller than required at the beginning of a process due to material constraints could prevent hardware from properly attaching to another component later on. Even if all the materials specified in a tutorial are on hand and are of the correct dimensions, if they are visually similar, it is also likely for the maker to need an enhanced level of material knowledge to distinguish between them. When this happens, the maker could use an alternative material that could have structural, health, or safety consequences that they do not know about because they are unfamiliar with the material (e.g., using polycarbonate instead of acrylic in a CO_2 laser cutter will generate chlorine gas). This can be further compounded if the instructions or tutorial they are following does not provide any material knowledge about, for example, material handling or appropriate disposal techniques [30]. Each of these material-specific challenges posed by explicit knowledge resources are due to a combination of low maker material literacy, inadequate descriptions and information provided in the instruction set, and the static nature of instructions themselves (i.e., a tutorial author cannot be expected to know which materials one has on hand or the level of knowledge of all makers who will consume their instructions). While literacy and descriptions can be improved via better graphic design and textual representations, it appears that solving these other challenges requires the design of dynamic and adaptable instruction sets.

Parts: Many of the same challenges that are imposed by the use of raw materials can also be found with the use of parts or components. When diagrams do not make it clear which parts are needed for a given step [24, 62, 82], makers do not have all the necessary parts on hand [62, 82], or they have the wrong type or model of parts [62, 82], makers are left to make substitutions, which could be inadequate or inappropriate. Additionally, if salient features of perceptually similar parts are not highlighted [58], for example, the differing lengths of two types of screws, there is an increased likelihood that a maker will use the wrong part but will not discover this until later in the process. Unlike with materials, the usage of incorrect or inadequate parts will force a maker to enter into a disassembly then reassembly phase, which could enable them to finish creating their artifact, however this also increases the likelihood that they introduce a potentially irreducible number of mistakes into their process [10]. The challenges that the inadequate labelling and identification of parts poses to fabrication and assembly speak to a larger need for systems that enable parts to be identified on a workbench or on a work-in-progress in real time such that if they are incorrectly utilized, the maker can quickly be made aware, using visualization techniques, that they are using the wrong part.

Tools: Similar to parts and materials, when a maker does not have access to a tool [56, 62, 82] or a set of instructions is unclear about which tool should be used [21, 82] or how it should be used [82], they will try to substitute one tool for another or use their best guess about what they should do. Unlike parts and materials, however, tools come with additional challenges. For example, even if a maker has a tool on hand and the instructions specify how it should be used, that tool may be in use by another maker or may be broken [56, 82]. In some cases, this could prevent a maker from being able to complete a task (e.g., they need to drill a hole to a specific diameter and the necessary drill bit is broken) or require them to substitute a different, possibly inappropriate tool. Another challenge that tools pose is that the maker may not have the necessary training or permission to use a tool that is required [82]. If a child were to download an Instructable to make a plastic birdfeeder[2] out of a plastic pop bottle, they would need to seek permission or the assistance of another older maker to help them cut a hole in the plastic bottle using the round file that is specified in the Instructable because this use of the file in this way is non-traditional and potentially dangerous. If this maker were to continue, they could potentially injure themselves or ruin their artifact due to the lack of tool literacy and training that they have. There could, of course, be systems created to automatically detect when a tool has been picked up and determine if it is appropriate to use in a given situation, however, a more fruitful approach could be to dynamically suggest alternative tools that the maker has the skills, knowledge, and permission to use.

Techniques: One last sub-category of equipment that poses challenges is the technique that instructions suggest a maker use. In some instances, an instruction set may not provide a technique to attach two components together, so a maker may be unaware of how to do so [24, 30]. In these cases, a maker must guess what they should do. The resulting action or method they choose could prevent future steps from being accomplished. Whenever explicit knowledge resources do not provide the necessary details to complete a given task, such as the working time of adhesives, ways material or excess need to be disposed of, or potential safety hazards [16, 30, 82], makers will be forced to seek external resources, thus decreasing their engagement with the task at hand and their trust in the resource they are using. Further to this, as many fabrication tutorials make use of new techniques that the author themselves has created, when the tutorial does not describe the design process or steps that lead to the technique, or the successes or failures that the author encountered along the way, it can be difficult for a maker to recreate or utilize the technique [77, 78, 79]. Alternatively, it is also common for instructions and tutorials to provide highly specialized details about a technique that require the specific properties of the tools, materials, or parts that the author had available [76]. This prevents a maker, for example, from being able to generalize a technique to a different brand of a tool (e.g., a plunge versus stationary router). In other instances, instructions may require multiple makers for a certain task but never indicate this until the task where a second maker is required [2]. This will lead to frustration and the artifact not being finished if one is working individually. Lastly, although only impactful for those who are left-handed, almost all instructions use diagrams and techniques that are illustrated from the right-handed perspective. For those makers who are left-handed, this can cause immense frustration as they must mentally flip the instruction they are viewing to match their handedness [30]. Many of the challenges that

---

[2] For example, the 1 Cap Water Bottle Bird Feeder: http://www.instructables.com/id/1-Cap-Water-Bottle-Bird-Feeder/.

techniques create, can be rooted in omissions or a misunderstanding on the part of the tutorial or instruction author has to the target audience of their document. As it is unlikely that a maker's context will be identical to an author's context, when differences occur that impact the techniques to be used, a maker can't simply look around their space and find the next best alternative. Instead, they must consult external resources, additional makers, or simply guess how to continue. There is thus a need to improve the tools that are available to designers to ensure that they are able to appropriately target their audience.

The deficiencies that arise within explicit knowledge resources due to the tools, materials, parts, and techniques that are to be used are varied; however, they largely relate to a lack of understanding about the target audience, the target audience's environment, skill set, material and tool knowledge, and so on. When makers are left to fill in the gaps themselves, they increase the likelihood for errors, safety issues, and the chances of not completing their artifact. Automatics' dynamicism and the tools it provides for designers and makers should improve the content that a designer creates and the content that a maker consumes.

3.2 Process

Another class of deficiencies relates to the prescribed task sequences and sequence flexibility inherent in the tasks to be performed.

Task Sequence: The order in which tasks are presented to makers can cause many issues. For example, when the order of a sequence of tasks is incorrect [16, 24, 25, 82], if the maker has already completed some tasks that were in the wrong order, they need to determine which tasks they should have done, determine the point to revert back to, undo the tasks if possible, and perform the tasks that should have been specified in the correct order to begin with. Provided that tasks can be undone, this increases task time and leads to frustration. When there are gaps between successive tasks, a maker is left to guess the tasks that they need to perform to get from task to task, which in some cases may lead to errors or require them to consult external resources (which will encourage more errors) [16, 24, 25, 82]. If a designer has improperly divided activities amongst tasks (e.g., five activities per task and then two activities per task) [24, 66, 82] or overloaded tasks with too many tasks [24, 44, 79], makers can forget to perform a task due to the sheer number of tasks they need to perform. Alternatively, when tasks are specified in such detail that they require many of the same repeated actions, makers may experience boredom and believe that they completed more tasks than they actually did [30, 44]. The manner in which a designer decides that a final artifact should be fabricated is also of concern. Given our natural tendency to want to assemble artifacts from the bottom to top, designers who organize tasks such that the tasks use alternative spatial patterns [44] may place undue cognitive load on makers, most of whom are already exerting great mental effort to overcome the other aforementioned issues. In addition to challenges with the sequence in which tasks are presented, when tutorials or instructions do not specify the complexity or duration required to complete each task, makers have no way to gauge and plan for the time it will take them to complete an artifact or how difficult it will be for them or others to complete a task or entire project [30, 82]. This could be at odds with the situational context of the maker and their goals. The challenges that the sequences of tasks makers are encouraged to follow highlight the need for authoring tools that are able to verify that sequences are in the correct order, do not require too many or few tasks at once, vary repeated tasks to discourage boredom, support our desires to build bottom to top, and provide estimates of task complexity and duration.

Sequence Flexibility: Like assembly instructions, where the goal is to assemble an artifact though a sequence of fixed tasks [4], current fabrication tutorials and instructions make use of a single pathway that is to be followed to completion [4, 19]. Explicit knowledge resources are unable to assist an maker when they make an error [19]; a maker must determine that they made an error, determine when this error occurred, undo the tasks they have taken since the error (if possible), correct the error, and re-preform a number of tasks to bring them back to where they were, and then continue with the task sequence. This increases the time they spend on a task and requires that a maker is able to determine that an error was made and how to fix it. Because resources are designed by one maker for another maker, resources are unable to accommodate drop-in makers who may want to participate for some, or all, of the fabrication process [4]. This rigidity goes against the very ethos of making and fabrication, which is to support serendipitous exploration and encourage learning via trial and error and collaboration with others. Lastly, the overhead that designers encounter when creating a single set of instructions often prevents them from creating sets of instructions tailored towards different skill levels or expertise [4, 30], reuse, remixing, or cultural variations [4, 53, 61, 62, 72], personalized variants [50, 71, 75, 76], or sustainable, artisanal, or cultural alternatives [66]. Each of these deficiencies underscore the importance of the next generation of instructions being able to adapt to the context at hand and the ever-changing goals and skill sets of makers. The static, rigid nature of task sequences found in instructions today are inadequate for the making population.

As the task sequence and task flexibility challenges identified thus far have demonstrated, if a maker is not provided with tasks that can be followed and the tasks cannot adapt to their context, workflow, or goals, they will encounter frustration and will often be unable to complete a task unless they seek out external resources or additional assistance. While the onus is on the designer to create task sequences that are free from mistakes and omissions, designers need support in the form of tools and simulations to create instruction sets that makers can follow. Dynamic Manuals attempt to solve many of these challenges through (i) the Constraint Specifier application, which uses sequence checking techniques, and (ii) the Maker Interface, a presentation mechanism that is able to adapt to the current environmental context to provide multiple pathways to reach the final, completed artifact.

3.3 Representation

The last category of deficiencies relates to the presentation of the tasks to be performed, i.e., the illustrations, text, and global features used.

Illustrations: Diagrams and figures are a crucial element of virtually all sets of instructions and tutorials, as they provide information that textual descriptions alone cannot. When diagrams are unclear or overloaded with details [2, 24, 30, 66] or do not highlight salient elements [44, 66], it can be difficult for a maker to understand what part of the diagram they should attend to. This could result in them not to noticing crucial details. Although many paper-based diagrams are presented in black, white, and grey, when the colors, symbols, and shading used to draw attention to specific aspects of a diagram are indistinguishable, it is easy for a maker to attend to the wrong aspect of a diagram or confuse certain parts of a figure with others. It is also common for a tutorial to use the incorrect type of diagram for illustration [2, 16, 24]. In such cases, instead of using an action diagram to show the direction and areas where components fit into a larger whole, an instruction may use, for example, a structural or explosion diagram, neither of which convey the action that needs to be performed to complete a given task [2]. Such mistakes make it difficult to understand the intent of the current task that should be performed. Lastly, the most common issue with diagrams is that the viewpoint of the diagram often changes between tasks, even if two tasks are on the same page [2, 16, 24, 30, 44, 66, 80]. When this occurs, makers cannot create a mental model of the artifact or understand how the component they are working with fits within the larger artifact. This only serves to add to the cognitive load the maker is already experiencing while fabricating and leads to breaks in task flow and maker annoyance.

Text: Although diagrams are staples of almost all instruction sets, text is often used to provide an additional level of detail that cannot be relayed quickly and easily using a figure. One common issue with the use of text is that text and diagrams are often unassociated [66], either on different pages or in different sections of the same page, and do not have callouts or references to each other. This creates a disconnect, requiring one to either annotate a task themselves with the required information or flip or scroll back and forth to gain a complete understanding of the task to perform. In some instances, verbose text passages are used in place of a more suitable diagram such as an action or explosion diagram [2, 16, 24], or provide detail that is unnecessary for the task at hand [66]. Such design choices may be the result of limited production budgets but end up overcomplicating the information being presented to a maker. As previously mentioned, because instructions are not able to adapt to the maker consuming them, text is also frequently written for the wrong audience or literacy level [22, 44, 78, 82] or in the wrong language [30]. Such issues increase the chance that a maker will use the wrong part, attach it in an incorrect place, or measure their material incorrectly because they are unable to read and comprehend the information provided to them. Lastly, as instruction sets are authored by a variety of different designers, there has yet to be a universal set of terms or domain-specific language used in the creation of instruction sets [82]. Because of this, every instruction or tutorial that is consumed by a maker differs in the way that processes and techniques are described. The lack of continuity and variation between tutorials makes it difficult for a novice maker to transfer the skills and knowledge they learn from one activity to another. Although text can enhance an illustration, these deficiencies demonstrate that it often hinders more than it can help. Designers thus need mechanisms to author instructions to suit a variety of makers and makers need the ability to dynamically modify, reformat, or reorganize instruction text to meet their skill level and literacy.

Global: Aside from issues with the text and illustrations that are provided, there are a number of global issues with explicit knowledge resources that pose challenges for makers. For example, stylistic issues, such as inconsistent formatting from task to task [30, 44, 66, 82], an inconsistent use of text and diagrams through a tutorial or set of instructions [30], and the use of a non-durable format (e.g., paper) [30], decrease the professionalism of resources and lessen the confidence a maker has in the tasks they are performing. Organizational issues including the inability to search through a set of instructions [30] or easily navigate back to a past task [4, 30] can prevent a maker from wanting to use other resources or search within a resource for answers because they may have difficulty finding their place again. The static nature of such resources also poses problems when one considers the dynamic nature of fabrication activities: instructions are not able to provide real-time assistance or advice [19], they cannot enable a maker to see what others have completed or the outstanding tasks left to perform [2], they cannot provide hints or suggestions based on other's makers experiences using the instructions [19, 71], and they are difficult to share amongst multiple makers all working on the same artifact [2]. In addition, it is common for pertinent details about the fabrication activity, such as the overall goal, to be missing [24, 30, 82]. This could be due to a lack of status checkpoints throughout the process [44], the lack of a motivating and goal-driven narrative throughout the tutorial [30], or the lack of clear details as to the end goal of each task that is to be performed [15, 82]. The omission of these details decreases beliefs that an artifact will eventually be completed. Lastly, as explicit knowledge resources offer single pathways to completion, they often fail to act as a reference or impart meaningful knowledge to the maker [30, 31]. This is because their goal is to illustrate one pathway to achieving a process, rather than encourage exploration or provide suggestions on areas where variation or spontaneity could be integrated. These global challenges speak to a more holistic need for the next generation of instruction sets to go further and do more than simply illustrating one path to completion – they need to be adaptable, dynamic mediums through which creativity and exploration can be supported and encouraged.

Some of these representation-related issues can be addressed by building better tools for tutorial and instruction authors that are able to compare an instruction set against a database of recommended diagrams and layouts to suggest improvements to the author. Others, however, require a community-wide agreement on a set of standards to make the transition between tasks and instruction sets easier for a maker to manage. Even with solutions to overcome organizational and continuity issues, each sub-category of challenges requires intelligence and awareness of the environment and maker(s) consuming the tasks. Without such knowledge, and a way to mediate and transition between different contexts, static instructions will not be able to adapt to multiple makers or tailor instructions to match a given maker's skill level or goals, all facets that will be required as fabrication activities continue to become more complex and collaborative.

3.4 Summary

The brief literature review presented above highlights the many challenges that are inherent in the design and consumption of explicit knowledge resources today. Although many of deficiencies arise from assembly activities, assembly is a crucial component of any fabrication task. Fabrication does, however, add an additional layer of complexity to the problems associated with assembly. For example, makers manipulate materials to obtain the 'parts' they need, such as cutting wood, shaping metal, molding clay, and so on. This increases the likelihood of mistakes (due to measurement errors, the use of offcuts, and the variety of tasks being performed). Unlike assembly, makers also often require many more tools to complete a task, some of which they may not have on hand (e.g., saw, clamps, orbital sander, etc.) or have knowledge about (e.g., bandsaw, laser cutter, casting materials, etc.).

While some challenges can be overcome by adopting visualizations or double-checking the information that is provided, others require additional research so that novel techniques can be developed. This work thus seeks to overcome some of these equipment, process, and representation challenges by using constraint satisfaction solvers, heuristic-based algorithms, parametric 3D models, wizard-style GUIs, interactive 3D models, the dynamic presentation of tasks, and information about the context within which fabrication is occurring. This allows present work to support makers in recovering from mistakes, utilizing alternative tools when needed ones are in use, forfeiting tasks if desired, and enabling multiple makers to participate in a fabrication activity, each with their own dynamically-generated set of tasks.

4 DYNAMIC GENERATION OF FABRICATION TASKS

Motivated by the challenges that current explicit knowledge resources present, the present work seeks to make the specification of fabrication and assembly tasks easier for designers, along with the consumption of said tasks by makers. The solution proposed in this work is to utilize a novel type of explicit knowledge resource that is dynamic and personalizable, i.e., it can adapt to a maker's current environmental context (e.g., the tools on hand, number of makers available, task preferences of a maker, and mistakes that were made). Before the architecture of these resources is described, the goals governing the design of the prototype system are presented. This is followed by a description of the modules and applications that constitute the conceptual architecture of a dynamic, explicit knowledge resource system.

4.1 Goals

Given the variety of deficiencies identified, there are many avenues along which fabrication task assignment and instruction presentation can be improved. Rather than attempt to solve every challenge, four opportunities directly related to fabrication workflows and the variety of challenges such contexts impose, were the focus of the present work.

G1: Dynamically Compute Fabrication Pathways

Inspired by the *sequence flexibility* and Task Sequence dimensions, fabrication tasks should be dynamically generated to support assembly activities (i.e., attach A to B), operational activities (e.g., drill A and paint B), and break-down activities (e.g., cut A into smaller pieces). By using constraints instead of fixed tasks, all possible ways an artifact can be fabricated should be known, thus permitting a reasonable pathway to be chosen from among the alternatives. Because tasks are generated by an algorithm that ensures all subcomponents are assigned to at least one task, and the constraints of the model are adhered to, forgetting to assign a task or assigning tasks in an inappropriate order should be impossible.

G2: Adapt to Uncertainty

Because explicit knowledge resources cannot adapt to the challenges that arise with the Equipment and Process dimensions, the prototype system should utilize information about maker preferences, mistakes, and tool status to determine which tasks to assign to each participating maker.

- *Forfeiting a task*: If a maker does not want to perform an assigned task (e.g., they are tired of sanding or desire to use other materials or equipment), other tasks that the maker could perform instead should be assigned.
- *Recovering from mistakes*: If a mistake is made, disassembly instructions should be generated such that the mistake can be corrected. If the mistake influences other aspects of the artifact (e.g., material cut too short), the artifact should be adjusted to allow for a variation of it to be created despite the mistake.
- *Adapting to the state of tools or preferences*: If a tool is being used by another maker, or cannot be used due to a lack of domain knowledge or a level of comfort, alternative tools should be suggested. If such tools are also unavailable, a different task that could be completed without said tool should be assigned instead.
- *Supporting design variants:* During fabrication, a maker may want to make a change to the artifact that they are creating, for example, make it longer, wider, or taller. The instructions that are provided should be able to dynamically change the visual and textual representations to accommodate design variations.

G3: Provide High-Level Global Context Awareness

As highlighted by the Global dimension, minimalism helps to reduce the cognitive load explicit knowledge resources impose [14]. It is, however, essential that enough information be provided for a maker to know where they are in the global context of the artifact [80]. Multiple ways to deduce global context should thus be provided and be accessible when appropriate, via dynamic, interactive methods.

G4: Support Drop-In Fabrication

Based on the *sequence flexibility* finding that current instructions do not accommodate multiple makers, support for individual and multiple makers should be provided via the automatic (re)division of tasks. If additional makers decide to drop in, tasks should be assigned such that a good task is chosen for a given maker, and a good maker is chosen for a given task.

4.2 Architecture

The architecture supporting the dynamic generation and assignment of tasks consists of three main elements (Figure 2). Artifact Applications are the applications used by a designer to create a digital representation of the artifact, i.e., a Parametric Modelling Application and a Constraint Specifier Application. The output resulting from the use of these two applications constitutes a Dynamic Manual, a novel explicit knowledge resource that contains all the data and information

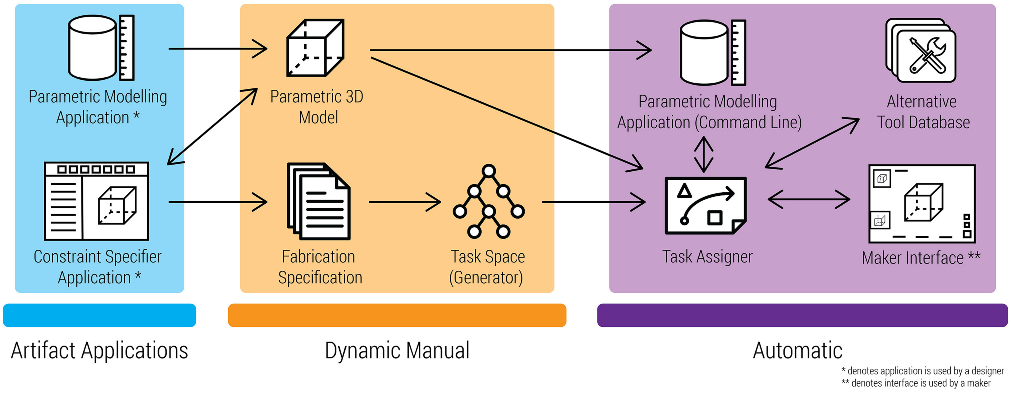


Fig. 2. The conceptual architecture necessary to support the dynamic generation of fabrication tasks. This architecture consists of *Artifact Applications* that are used by a designer to create digital representations of an artifact (i.e., a Parametric Modelling Application and a Constraint Specifier Application). These digital representations form a *Dynamic Manual*, i.e., a dynamic, explicit knowledge resource. It also includes an *Automatic*, which uses the content of the Dynamic Manual to assign fabrication tasks to makers via the Task Assigner, and visualizes tasks for a maker and collects information about the current environmental context from the maker via the Maker Interface.

necessary for the dynamic generation and assignment of maker tasks. The Dynamic Manual contains the Fabrication Specification and Parametric 3D Model authored by a designer and uses these components to generate a Task Space of all possible ways the artifact can be created within the constraints of the artifact's components. The Dynamic Manuals' Task Space is then used by an Automatic. An Automatic is a prototype system that uses the Task Space to assign tasks to makers via the Task Assigner, present tasks to makers and collect contextual information from them via the Maker Interface, and, if necessary, adapt the current artifact to handle personalization and error recovery (via the Command-Line Parametric Modelling Application) or provide alternative tool suggestions (via the Alternative Tool Database). Thus, it is these three main elements, the Artifact Applications, the Dynamic Manual, and an Automatic, that enable tasks to be dynamically generated, assigned, and visualized for makers.

4.2.1 Parametric Modelling Application and Parametric 3D Model. The basis of all artifacts that are supported is a parametric 3D model. Parametric models use parameter-based attributes to specify the geometric properties of a subcomponent (e.g., width, length, depth, and orientation). As such, a change to one parameter can be propagated through an entire model, thus, maintaining design intent during future modifications to the model. Such models enable an Automatic to provide support for error recovery, as a model can be regenerated if fabrication errors occur (e.g., a subcomponent was cut too small) or if personalization is desired (e.g., a subcomponent is made thicker for stability reasons).

The current implementation of an Automatic uses the open-source FreeCAD software (version 0.16) to enable artifacts to be modelled by a designer using a GUI (i.e., Figure 2 – Parametric Modelling Application). It also has support for Python-based, command-line access to the software (i.e., Figure 2 – Parametric Modelling Application (Command Line)). This command-line access enables an Automatic to update the geometric properties of the artifact model due to maker fabrication errors or personalization choices, and regenerate the underlying model at runtime. Once a model has been regenerated, all future tasks will use the new geometric properties of the model. For example, if a maker cuts a piece of wood too short, they can indicate the dimensions of the erroneous piece of wood and the Automatic will scale the associated model to match the new length and indicate the changes that need to be made to the rest of the subcomponents in the artifact, if necessary (G2). The parametric models created by a designer and the FreeCAD back-end thus enable a maker to continue fabricating their artifact while the Automatic takes into account their design choices or fabrication actions, in a seamless and invisible manner.

4.2.2 Constraint Specifier Application and Fabrication Specification. A necessary aspect of any explicit knowledge resource is the specification of the assembly constraints, material manipulations, and tools needed to create an artifact. Although point-and-click methods of interaction have been used to specify assembly constraints (e.g., FreeCAD Assembly2 workbench[3] and Autodesk's Inventor[4]) and prior work has explored automated methods to infer assembly constraints from 3D models [32, 56], human motion sequences [33, 46, 92], static 2D instruction sets [47], and cognitive engineering models [1], such techniques do not support the manipulation of materials, components that are derived from subcomponents, or allow for the assignment of tools. Instead of requiring a designer to manually author a text document that contains such information, the Constraint Specifier application was developed. It enables a designer to not only specify assembly constraint information, but also component derivations, material manipulations, tool usage, and so on, in a wizard-style manner. The specification of attachments between components is modelled after FreeCAD and Inventor's point and click techniques, which allow a designer to click on different faces in a model to indicate that the faces are constrained to each other. Note that all other information that is specified by a designer is provided via (pre-populated) dropdown boxes, text fields, or by selecting different components in the model.

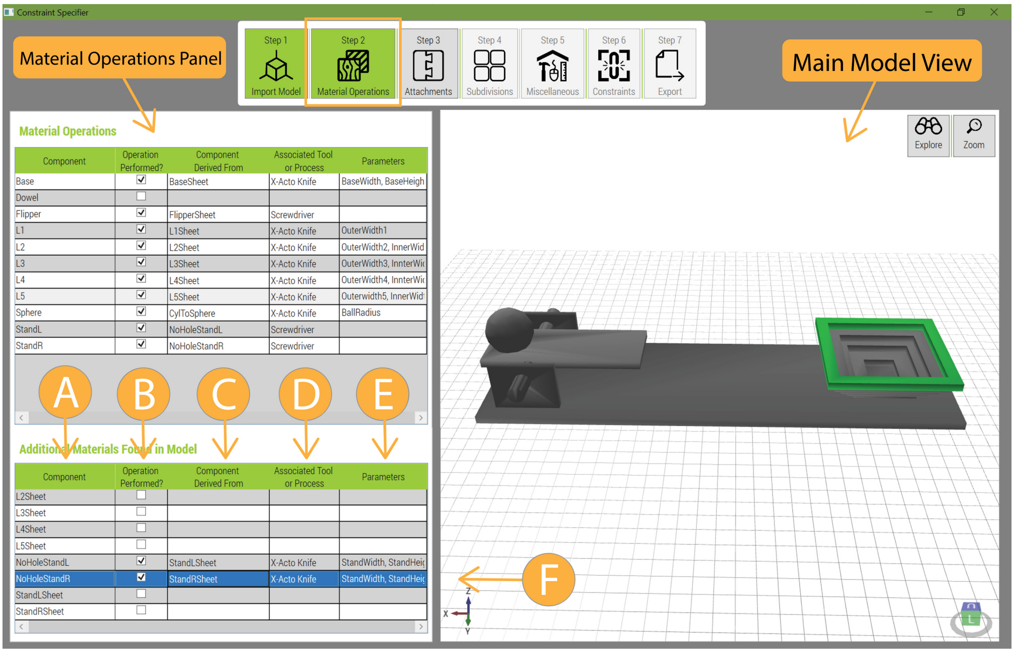
Using the application, a designer can load the parametric 3D model corresponding to their artifact and view it in the Main Model View on the right side (Figure 3). The designer can then use the list of model components in the Material Operations Panel on the left side (Figure 3(A)), which was populated with information from the 3D model to indicate (i) if a subcomponent has is derived from another component or raw material (i.e., Figure 3(B, C)), (ii) the tool or process used in the derivation operation (i.e., Figure 3(D); e.g., a utility knife to cut a subcomponent from a larger sheet of material, as in Figure 3(F)), and (iii) specify the labels for any maker-facing parameters that are required to create the component (i.e., Figure 3(E); the width and height of the stand, as in Figure 3(F)). The specification (or lack of specification) of these four types of information are used by Automatics to generate a portion of the space of all possible tasks that must be completed.

Once the details about the materials and material operations have been specified, the designer can then use the Attachments Panel to specify all the connections that exist between subcomponents (Figure 4). This is done by pressing or touching the "Select Components" button to switch the interface into selection mode, selecting the components in the Main Model View, and creating an attachment using the "Add Attachment" button. This series of actions will add a new row to the list on the left, populate it with the components that were selected (Figure 4(A, B)), and subsequently allow the designer to provide the information that the Automatics' Maker Interface and Task Assigner need to know about the nature of the attachment (Figure 4(F)), i.e., (i) which of the two components should be animated in the Maker Interface (Figure 4(C)), (ii) the type of animation that should be used (Figure 4(D)), and (iii) the tool that is required to create the connection (i.e., Figure 4(E); e.g., a hot glue gun to glue L4 to L5 as in Figure 4). The animation and specification of which component moves or rotates are used by the Maker Interface to determine the type and

---

[3] https://www.freecadweb.org/wiki/Sandbox:Assembly2.

[4] http://autode.sk/2mYaIZh.



Material Operations

Component	Operation Performed?	Component Derived From	Associated Tool or Process	Parameters
Base	☑	BaseSheet	X-Acto Knife	BaseWidth, BaseHeigh
Dowel	☐			
Flipper	☑	FlipperSheet	Screwdriver	
L1	☑	L1Sheet	X-Acto Knife	OuterWidth1
L2	☑	L2Sheet	X-Acto Knife	OuterWidth2, InnerWid
L3	☑	L3Sheet	X-Acto Knife	OuterWidth3, InnterWi
L4	☑	L4Sheet	X-Acto Knife	OuterWidth4, InnterWi
L5	☑	L5Sheet	X-Acto Knife	Outerwidth5, InnerWid
Sphere	☑	CylToSphere	X-Acto Knife	BallRadius
StandL	☑	NoHoleStandL	Screwdriver	
StandR	☑	NoHoleStandR	Screwdriver	

Additional Materials Found in Model

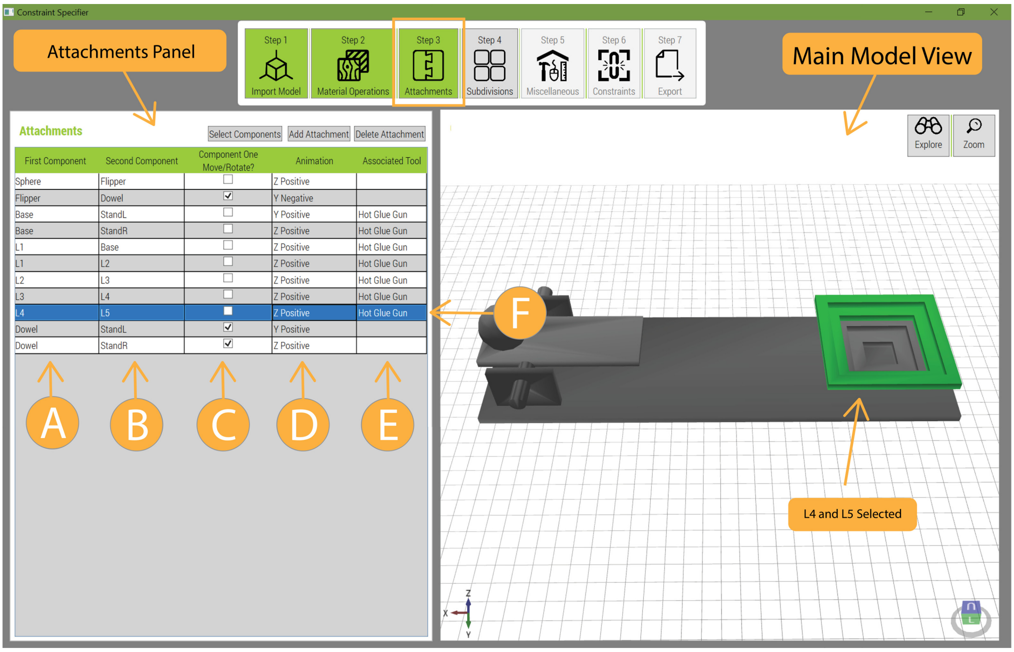
Component	Operation Performed?	Component Derived From	Associated Tool or Process	Parameters
L2Sheet	☐			
L3Sheet	☐			
L4Sheet	☐			
L5Sheet	☐			
NoHoleStandL	☑	StandLSheet	X-Acto Knife	StandWidth, StandHei
NoHoleStandR	☑	StandRSheet	X-Acto Knife	StandWidth, StandHei
StandLSheet	☐			
StandRSheet	☐			

Fig. 3. In the Constraint Specifier application, a designer can use the Material Operations Panel to indicate how subcomponents are created (or the other materials that they are derived from) by filling the necessary details into the rows of the lists on the left (B–E). The data for the Component column (A) is pre-populated from the loaded parametric 3D model, which can be manipulated on the right in the Main Model View. This example, in addition to the examples in Figures 4–7, demonstrates how a designer would specify the parameters and information necessary to create the ball catapult game for the User Study in Section 5.

direction of animations that should be used when the task is presented to a maker. The associated tool information is used by the Maker Interface to display the correct operation information and by the Task Assigner when determining which task to assign to a maker.

If a designer wishes to decrease the need to switch between tools during a task or pass materials back and forth between participating makers, they can use the Subdivisions Panel to specify which subcomponents, if any, should be assembled or modified at the same time, or be assembled or modified by the same maker (Figure 5). This is done by creating groups (i.e., subcomponents that should be assembled by the same maker) and subdivisions (i.e., subcomponents that should be assembled as a unit at the same time). The process to create each of these is the same. First, the maker presses or clicks on the “Select Components” button. Then they can click or touch all the components in the Main Model View that should be part of the group or subdivision. Once done, they click or press on the “Add Group” or “Add Subdivision” button and the group or subdivision will be added to the list on the left (Figure 5(A)). This information is used by the Task Assigner when determining which task a maker should be assigned next.

Once the designer has created groups and subdivisions (if any), they can, then, use the Miscellaneous Panel to specify parameters and unit information about the various components that make up the model (Figure 6). As the parametric 3D model uses variables and parameters to regenerate the model, for every row in the list on the left side (Figure 6(E)), the designer must specify (i) the parametric model parameters that correspond to the maker-facing labels input during in the Material Operations Panel (i.e., Figure 6(A, B)), (ii) the units that dictate the measurements of



First Component	Second Component	Component One Move/Rotate?	Animation	Associated Tool
Sphere	Flipper	☐	Z Positive	
Flipper	Dowel	☑	Y Negative	
Base	StandL	☐	Y Positive	Hot Glue Gun
Base	StandR	☐	Z Positive	Hot Glue Gun
L1	Base	☐	Z Positive	Hot Glue Gun
L1	L2	☐	Z Positive	Hot Glue Gun
L2	L3	☐	Z Positive	Hot Glue Gun
L3	L4	☐	Z Positive	Hot Glue Gun
L4	L5	☐	Z Positive	Hot Glue Gun
Dowel	StandL	☑	Y Positive	
Dowel	StandR	☑	Z Positive	

Fig. 4. The Constraint Specifier enables a designer to indicate which subcomponents are attached to which others in the Attachment Panel. The designer can select the subcomponents in the Main Model View and add an attachment using the “Add Attachment” button. This action will add a new row to the list in the Attachment Panel on the left side (A, B, F). The designer can further populate this row with information about which subcomponent would be animated (if any; C), the type of animation for the attachment (if any; D), and the tool that is used (E), via dropdown boxes.

the operation to be performed on the subcomponent (i.e., Figure 6(C)), and (iii) the component(s) that are associated to that parameter (Figure 6(D)). The parametric model parameters and associated information are used by the Maker Interface to map the maker-facing labels to the underlying parameters of the model so the correct label can be shown for a given task. The Maker Interface also uses the units to provide the correct measurements to a maker on a given task.

Lastly, the Constraint Panel enables the designer to use a tile-based interface to build simple Boolean expressions to specify any assembly constraints that exist in the artifact, for example, part A before part B, part B after part C, part A and part B before part C, and so on (Figure 7). To create a constraint, the designer drags the appropriate constraint tile (Figure 7(A)) into the grey Constraint Operations area and then uses the pre-populated dropdown boxes to specify the subcomponent attachments (Figure 7(B)). Once this is done, the designer can press or click on the “Add Constraint” button to have the constraint added to the Constraint List (Figure 7(C)). Note that the pre-populated data comes from the information specified by the designer in the Attachment Panel. The logic for the constraints does not have to be entered in any particular order because the Task Assigner will use each constraint to automatically generate the task order at run-time based on the current environmental context.

Once all constraint information has been input, the designer can click on the “Export” button. The Constraint Specifier then iterates through all the information that the designer has input to ensure that every subcomponent has the necessary information (i.e., all subcomponents are attached to at least one other subcomponent, each derived subcomponent has a tool and material

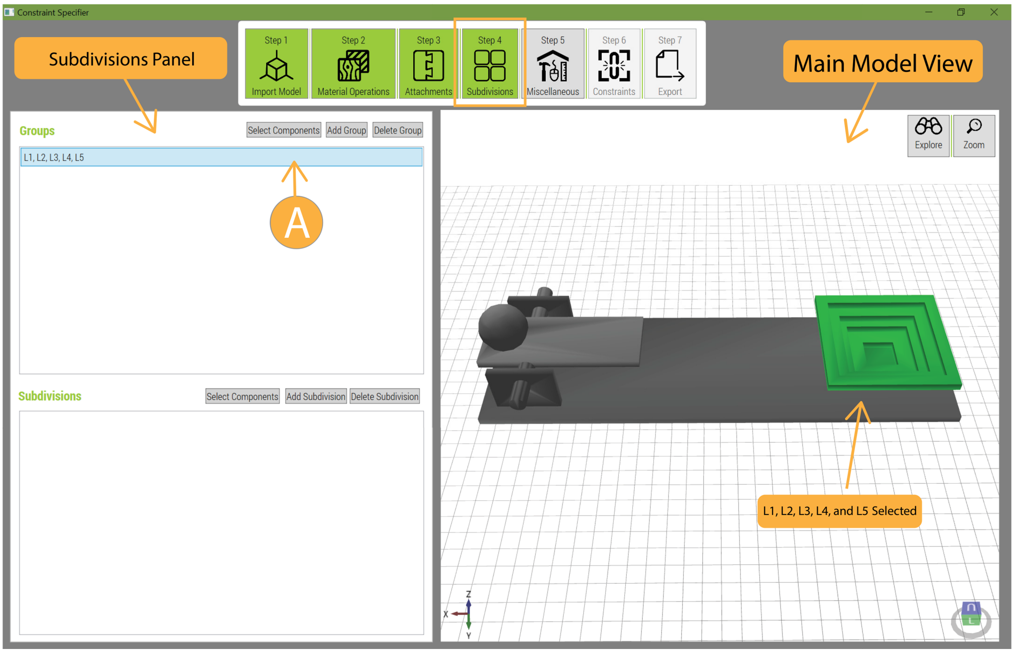
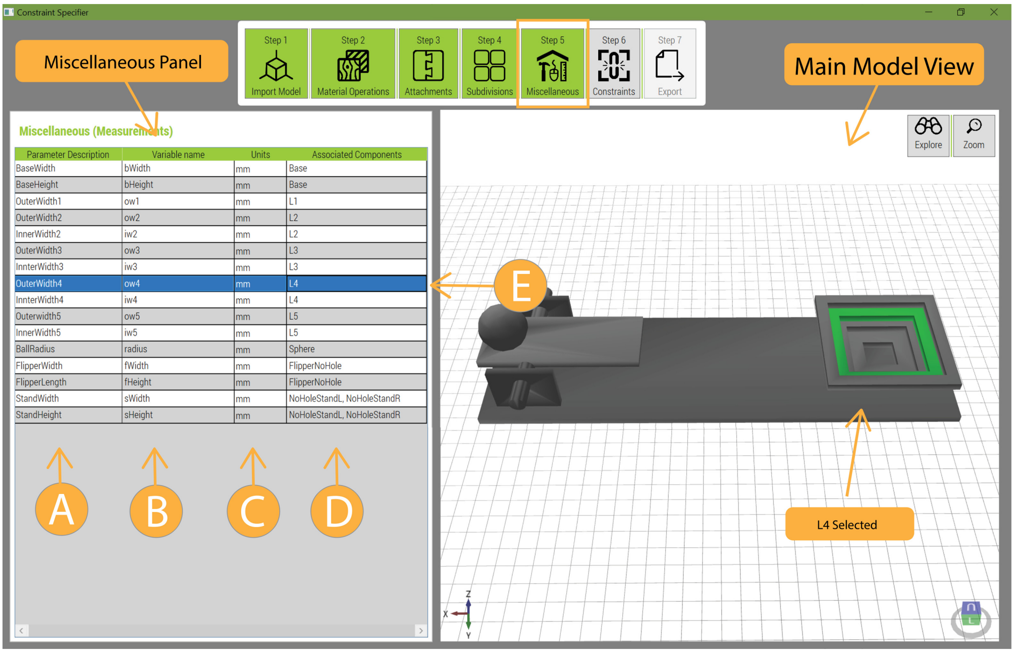


Fig. 5. As some subcomponents can be combined to form larger logical components, or others may be best created by a single maker, the Constraint Specifier enables a designer to indicate which subcomponents should be attached to others by the same maker, by selecting them in the Main Model View and creating a new group (A). For those subcomponents where a designer would prefer them to be completed at the same time, a designer can select the subcomponents in the main view and create a new subdivision.

specification and units, every maker-facing parameter is mapped to a variable name and every maker-facing parameter is association to at least one subcomponent, etc.). If there is any information that has not been specified, for example, not all components have at least one attachment, attachments are missing tools, or attachments do not have any constraints specified, among others, the application will open a dialog alerting the designer to the details that are missing and need to be added. The application will then ask the designer to go back into the interface and provide the missing details. Provided all pertinent details were specified, the application will output a text and JSON-based Fabrication Specification file (Figure 8). The file uses a simple, human-readable format to specify the attachments, assembly constraints, subdivisions and groups, material operations, and miscellaneous parameters that were specified by the designer in each panel of the Constraint Specifier application. An Automatic uses this information generate animations and visual content within the Maker Interface. This file is also used to create the Task Space and is used by the Task Assigner to assign tasks to makers. An example of the entire file that is used to specify the constraints and fabrication information for the ball catapult game used in the User Study in Section 5, is provided in Figure 8.

As the Constraint Specifier was created to negate the need to manually type out fabrication-based information for use with assembly constraints, it is novel and proposes one possible solution to fill the void left by current software tools and techniques. However, it uses existing user input techniques to convey assembly constraint information (i.e., point and click techniques similar to FreeCAD and Inventor) and combines them with standard methods to specify, gather, and verify



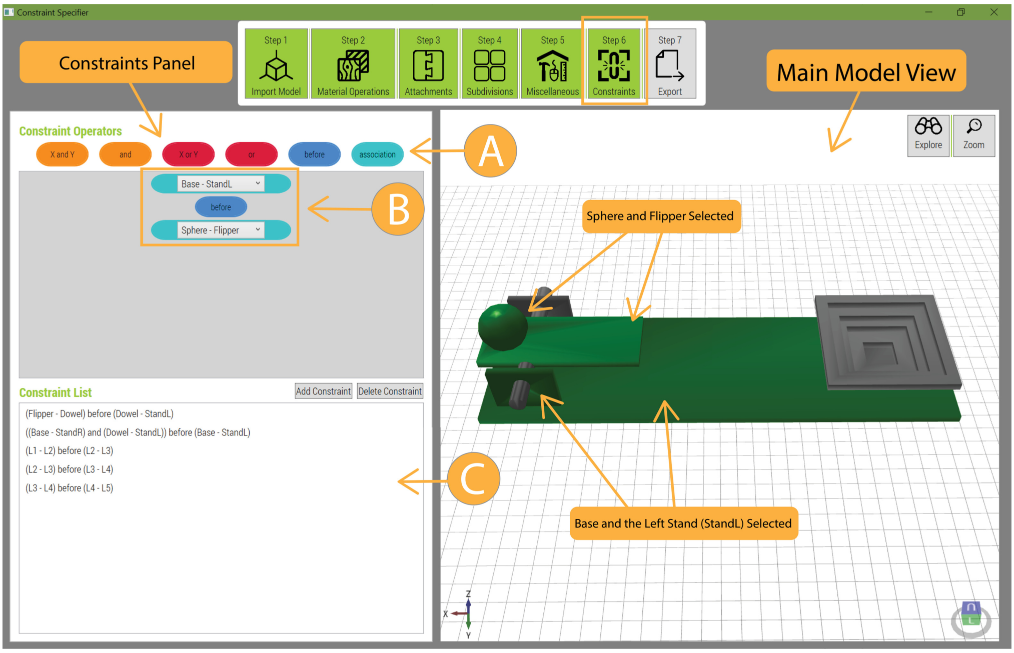
Miscellaneous (Measurements)

Parameter Description	Variable name	Units	Associated Components
BaseWidth	bWidth	mm	Base
BaseHeight	bHeight	mm	Base
OuterWidth1	ow1	mm	L1
OuterWidth2	ow2	mm	L2
InnerWidth2	iw2	mm	L2
OuterWidth3	ow3	mm	L3
InnterWidth3	iw3	mm	L3
OuterWidth4	ow4	mm	L4
InnterWidth4	iw4	mm	L4
Outerwidth5	ow5	mm	L5
InnerWidth5	iw5	mm	L5
BallRadius	radius	mm	Sphere
FlipperWidth	fWidth	mm	FlipperNoHole
FlipperLength	fHeight	mm	FlipperNoHole
StandWidth	sWidth	mm	NoHoleStandL, NoHoleStandR
StandHeight	sHeight	mm	NoHoleStandL, NoHoleStandR

Fig. 6. As parametric models utilize parameters for the specification of geometric relationships, and some parameters may apply to more than one subcomponent, the Miscellaneous Panel of the Constraint Specifier application allows a designer to indicate which maker-facing labels correspond to which parametric model parameters (A, B), the units of these parameters (C), and the subcomponents that utilize this parameter (D). The subcomponents associated to the given row are highlighted in the model on the right.

gathered information (i.e., using text boxes and pre-populated dropdowns, highlighting selected components in a 3D model, etc.). As the Constraint Specifier essentially duplicates the information that could be specified manually by a designer (with the addition of interactive elements, methods to refer to information from one section in another section, and the ability to verify the data that was input), a user study was not run to evaluate the utility of the Constraint Specifier. Such an evaluation would be fruitful to explore if techniques to automatically infer constraint-based information were integrated within the Constraint Specifier and these techniques required the author to confirm that the inferred constraints were correct.

4.2.3 Task Space (Generator) and Task Assigner. The Task Space determines the order tasks need to be completed (i.e., the order of connections between subcomponents) using a Python-based generator that is aware of the physical constraints of the model via the Fabrication Specification file. It uses the method specified by De Mello and Sanderson [27] and requires the specification of two types of constraints for the derivation: precedence constraints and operation constraints. Precedence constraints indicate, for example, that connection A–B must be formed before connection B–C. Such constraints are represented as a Boolean expression of connections. For example, ((A–B || A–C) & A–D) > D–E indicates that subcomponent A must be connected to subcomponent D and either subcomponent B or C, before subcomponent D can be connected to E. Operation constraints, on the other hand, convey that certain groups of components should be treated as a whole, for example, each of the four screws holding on a lock should be put in one after another, but not in a specific order. They can be represented as a unified component, for example, "{A, B,

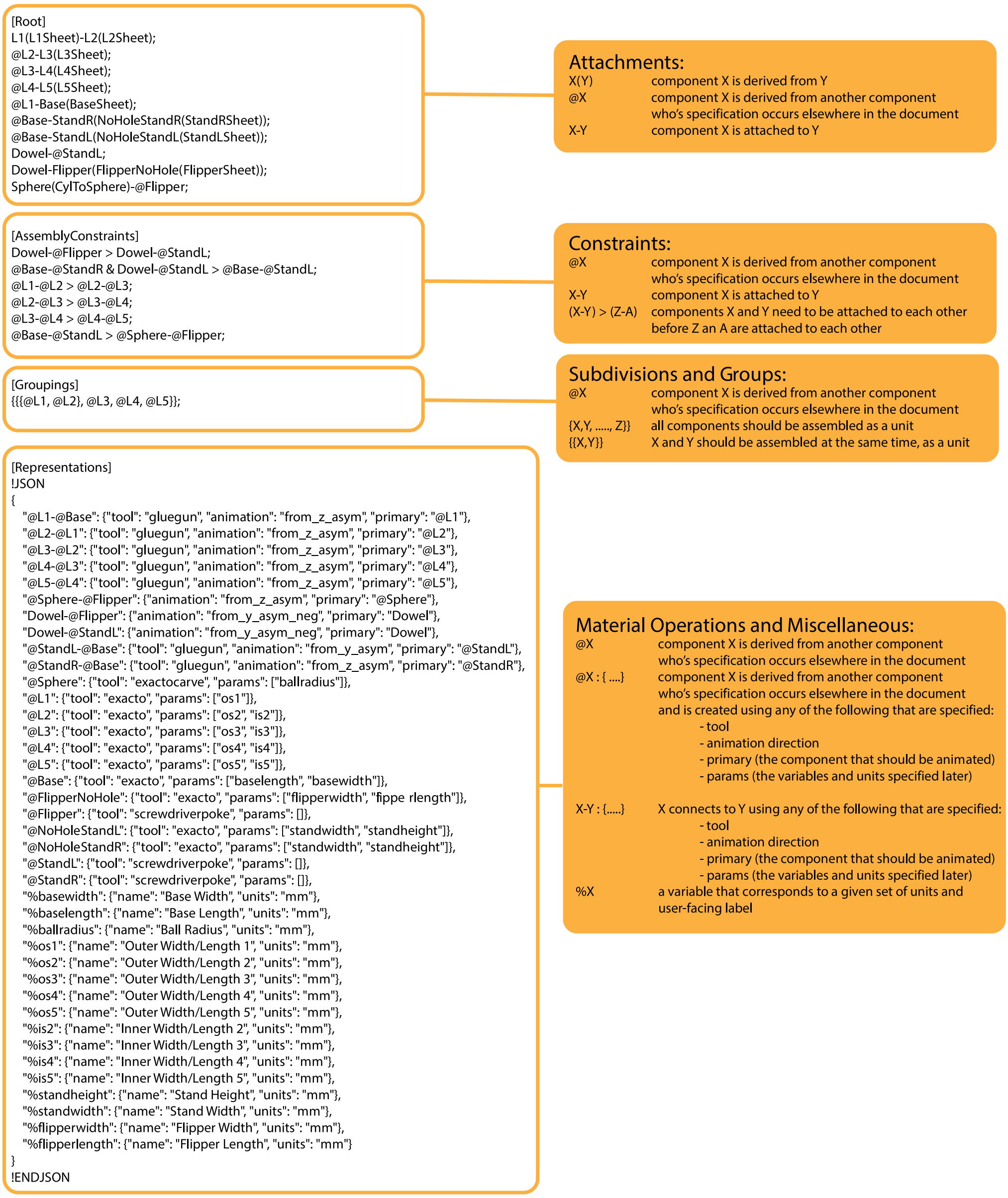


(Flipper - Dowel) before (Dowel - StandL)
((Base - StandR) and (Dowel - StandL)) before (Base - StandL)
(L1 - L2) before (L2 - L3)
(L2 - L3) before (L3 - L4)
(L3 - L4) before (L4 - L5)

Fig. 7. To specify the assembly constraints for an artifact, the designer can use the Constraints Panel to build simple Boolean statements using the constraint operator tiles on the left (A). The dropdown box contained within the “X” or “Y” tiles can be used to specify meshes in the model (the associated subcomponents will be highlighted in the model on the right; B). Once each statement is complete, it can be added to the Constraint List (C), where it is converted into a traditional Boolean statement.

C, D}”, where A–D are all individual subcomponents that should be operated on at the same time; however, the specific order that they are operated on does not matter (e.g., A can be attached, then B, C, and finally D, or B can be attached then D, C, and A). The generator uses both types of constraints to create an AND/OR graph of all possible trees representing every possible way an artifact can be fabricated (i.e., the Task Space; G1). Each node in the graph represents a possible state of completion, or task, of the final artifact. Due to the binary nature of the AND/OR graph, it is assumed that every connection involves exactly two constituent subcomponents, so each task makes use of at most two components (i.e., three components will not be utilized for a given task). As the Task Space is generated using the method described by De Mello and Sanderson in [27], the method itself is not novel. The application of their method to a fabrication, as opposed to assembly, scenario, and its use within an interactive and dynamic system such as an Automatic is, however, novel. The reader is referred to [27] for the correctness and completeness proof De Mello and Sanderson’s method.

The generator uses a divide-and-conquer approach. Starting with the completed artifact, the generator takes the entire artifact and removes one subcomponent or operation (e.g., cut A from material B), splitting the artifact into two subcomponents. It then checks if the constraints of each subcomponent are satisfied. If they are, this task could be assigned to a maker. The system then recursively removes and splits the artifact until all possible ways the artifact could be created have been found, and repeats this process for each subcomponent and operation. The result of this process is an AND/OR graph that is composed of smaller AND/OR subgraphs that represent



```
[Root]
L1(L1Sheet)-L2(L2Sheet);
@L2-L3(L3Sheet);
@L3-L4(L4Sheet);
@L4-L5(L5Sheet);
@L1-Base(BaseSheet);
@Base-StandR(NoHoleStandR(StandRSheet));
@Base-StandL(NoHoleStandL(StandLSheet));
Dowel-@StandL;
Dowel-Flipper(FlipperNoHole(FlipperSheet));
Sphere(CylToSphere)-@Flipper;
```

```
[AssemblyConstraints]
Dowel-@Flipper > Dowel-@StandL;
@Base-@StandR & Dowel-@StandL > @Base-@StandL;
@L1-@L2 > @L2-@L3;
@L2-@L3 > @L3-@L4;
@L3-@L4 > @L4-@L5;
@Base-@StandL > @Sphere-@Flipper;
```

```
[Groupings]
{{{@L1, @L2}, @L3, @L4, @L5}};
```

```
[Representations]
!JSON
{
   "@L1-@Base": {"tool": "gluegun", "animation": "from_z_asym", "primary": "@L1"},
   "@L2-@L1": {"tool": "gluegun", "animation": "from_z_asym", "primary": "@L2"},
   "@L3-@L2": {"tool": "gluegun", "animation": "from_z_asym", "primary": "@L3"},
   "@L4-@L3": {"tool": "gluegun", "animation": "from_z_asym", "primary": "@L4"},
   "@L5-@L4": {"tool": "gluegun", "animation": "from_z_asym", "primary": "@L5"},
   "@Sphere-@Flipper": {"animation": "from_z_asym", "primary": "@Sphere"},
   "Dowel-@Flipper": {"animation": "from_y_asym_neg", "primary": "Dowel"},
   "Dowel-@StandL": {"animation": "from_y_asym_neg", "primary": "Dowel"},
   "@StandL-@Base": {"tool": "gluegun", "animation": "from_y_asym", "primary": "@StandL"},
   "@StandR-@Base": {"tool": "gluegun", "animation": "from_z_asym", "primary": "@StandR"},
   "@Sphere": {"tool": "exactocarve", "params": ["ballradius"]},
   "@L1": {"tool": "exacto", "params": ["os1"]},
   "@L2": {"tool": "exacto", "params": ["os2", "is2"]},
   "@L3": {"tool": "exacto", "params": ["os3", "is3"]},
   "@L4": {"tool": "exacto", "params": ["os4", "is4"]},
   "@L5": {"tool": "exacto", "params": ["os5", "is5"]},
   "@Base": {"tool": "exacto", "params": ["baselength", "basewidth"]},
   "@FlipperNoHole": {"tool": "exacto", "params": ["flipperwidth", "fippe rlength"]},
   "@Flipper": {"tool": "screwdriverpoke", "params": []},
   "@NoHoleStandL": {"tool": "exacto", "params": ["standwidth", "standheight"]},
   "@NoHoleStandR": {"tool": "exacto", "params": ["standwidth", "standheight"]},
   "@StandL": {"tool": "screwdriverpoke", "params": []},
   "@StandR": {"tool": "screwdriverpoke", "params": []},
   "%basewidth": {"name": "Base Width", "units": "mm"},
   "%baselength": {"name": "Base Length", "units": "mm"},
   "%ballradius": {"name": "Ball Radius", "units": "mm"},
   "%os1": {"name": "Outer Width/Length 1", "units": "mm"},
   "%os2": {"name": "Outer Width/Length 2", "units": "mm"},
   "%os3": {"name": "Outer Width/Length 3", "units": "mm"},
   "%os4": {"name": "Outer Width/Length 4", "units": "mm"},
   "%os5": {"name": "Outer Width/Length 5", "units": "mm"},
   "%is2": {"name": "Inner Width/Length 2", "units": "mm"},
   "%is3": {"name": "Inner Width/Length 3", "units": "mm"},
   "%is4": {"name": "Inner Width/Length 4", "units": "mm"},
   "%is5": {"name": "Inner Width/Length 5", "units": "mm"},
   "%standheight": {"name": "Stand Height", "units": "mm"},
   "%standwidth": {"name": "Stand Width", "units": "mm"},
   "%flipperwidth": {"name": "Flipper Width", "units": "mm"},
   "%flipperlength": {"name": "Flipper Length", "units": "mm"}
}
!ENDJSON
```

Attachments:

X(Y)	component X is derived from Y
@X	component X is derived from another component who's specification occurs elsewhere in the document
X-Y	component X is attached to Y

Constraints:

@X	component X is derived from another component who's specification occurs elsewhere in the document
X-Y	component X is attached to Y
(X-Y) > (Z-A)	components X and Y need to be attached to each other before Z an A are attached to each other

Subdivisions and Groups:

@X	component X is derived from another component who's specification occurs elsewhere in the document
{X,Y, ..., Z}}	all components should be assembled as a unit
{{X,Y}}	X and Y should be assembled at the same time, as a unit

Material Operations and Miscellaneous:

@X	component X is derived from another component who's specification occurs elsewhere in the document
@X : { ....}	component X is derived from another component who's specification occurs elsewhere in the document and is created using any of the following that are specified: - tool - animation direction - primary (the component that should be animated) - params (the variables and units specified later)
X-Y : {.....}	X connects to Y using any of the following that are specified: - tool - animation direction - primary (the component that should be animated) - params (the variables and units specified later)
%X	a variable that corresponds to a given set of units and user-facing label

Fig. 8. The Fabrication Specification file that is the result of the Constraint Specifier Application and is used by the Task Assigner. Each segment of the file is derived from the information that the designer inputs into the application, however, the designer does not require any specific technical knowledge to do so, because the Constraint Specifier translates the information they input into the text fields and dropdown boxes into the format that an Automatic needs to create the Task Space and generate each task.

every possible way all the subcomponents can be fabricated (i.e., the Task Space). As operations are explicitly defined to act on only one subcomponent, they are treated as independent tasks, with a single child task representing the component that they act on.

The Task Assigner is written in Python and uses the Flask Framework to allow the Maker Interface to communicate with the Task Space, Alternative Tool Database, and the command-line Parametric Modelling Application. Much like GPS recalculation, the Task Assigner traverses the associated Task Space graph to assign tasks to makers in real time (G1). Every time a maker requests a task, the Task Assigner uses a greedy approach to traverse all remaining potential tasks within the Task Space, assigning a "possibility" score to each one. Currently, the possibility score is a weighted sum of four heuristics; however, other heuristics could be integrated to support, for example, a maker's skill level or equipment certifications.

- *Opt out score*: As an Automatic enables a maker to opt out of using certain tools or performing certain tasks, the Task Assigner tracks each time a tool or task is refused. The current implementation sums the number of times the potential task was refused and the number of times the tool for the task was unavailable. The result of this computation is the deprioritization of that tasks that the maker has already deferred or tasks where tools were unavailable.
- *Disassembly score*: Because an Automatic can support the disassembly of subcomponents if they were attached in error, if a maker is currently in a disassembly state, they should continue to disassemble subcomponents – not transfer disassembly tasks to another maker. As such, the disassembly score determines how many disassembly tasks are remaining for the maker by traversing from each potential subgraph to the subgraph that contains the task where the error was performed and computing the distance between the graphs. If the maker is not disassembling, the disassembly score is "1." The result of this score ensures that the maker who made the mistake is the one who fixes it, and the shortest path is taken to complete the disassembly tasks required to correct the error.
- *Tool continuation score*: An Automatic prefers to keep assigning tasks to a maker that use the same tool that they were just working with. This prevents makers from having to continually switch between tools, and from tools potentially becoming unavailable because they are in limited availability or are in use. In the current implementation, a score of "1" was assigned if the potential subcomponent required the same tool or "0.5" if it did not require a tool or required a tool different than the last subcomponent. This measure could be extended to utilize an Automatic's database of alternative tool suggestions and compute a more fine-grained tool similarity score in the future.
- *Component similarity score*: If a maker was just working on with a component or subcomponent, they should continue to work on other related subcomponents that are part of the component. This will minimize the cost of switching between tasks that are unrelated or having to pass subcomponents back and forth between other participating makers. The current implementation determines how many subcomponents that are part of the current task are also contained within a potential task and uses this value as the score.

Although all four scores are used in the computation of the possibility score, the disassembly score and opt out scores are weighted an order of magnitude higher than the tool continuation and component similarity scores to ensure that potentially confusing and frustrating tasks are presented less often than those which are based on preferences. Also, note that if only one maker is participating, the opt out score becomes less important, because until another maker joins, the current maker is the only one who can complete the task. In this case, any potential tasks that could be assigned are deferred until all other tasks not requiring the subcomponents within the currently rejected task have been completed. At this point, provided no other makers have joined, there is no choice but to assign the tasks to the current maker.

Once all tasks have been assigned a possibility score, the Task Assigner then chooses the task with the minimum score and assigns it to the maker. This process repeats until there are no more tasks that can be assigned to that maker. If a maker is not satisfied with their assigned task, they can forfeit it (i.e., the remaining task will be assigned to other makers) and request the next best task for them (i.e., the task with the next lowest possibility score). Once all tasks that have not been forfeit have been assigned, the maker will then be presented with the next best task from the tasks that remain from the forfeited ones, provided that no other makers have joined the activity. In some cases, this may result in the maker performing a series of tasks they did not want to perform in succession, or in other cases, these tasks may be interleaved amongst other non-forfeit tasks due to the constraints inherent in the artifact. If other makers have joined since the task was forfeit, said task may be assigned and completed by the other maker(s). Due to the level of uncertainty inherent in fabrication processes (i.e., tools may become broken at any time, makers may leave or join the activity, a mistake may be made, etc.), the current implementation optimizes to find the best task for each maker at the task assignment time. This helps ensure that maker frustration can be mitigated, at least in the short term, by giving a maker something else to do when they do not want to complete a task, thus, enabling them to still feel as if progress can be made on their artifact. There are, of course, other heuristics that could be applied to such a greedy approach to, for example, redistributing forfeit tasks throughout the rest of the making activity, however, this would need to be paired with visual feedback and explanations in the Maker Interface so that the maker could understand why, even if they wanted to forfeit a task, they must perform it instead of forfeiting it, or perform it soon after they forfeited it.

The Task Assigner is not interested in the number of makers working on an artifact, only which tasks have been completed or are in progress. As such, there is no limit to the number of makers who can participate. When multiple makers could be assigned tasks simultaneously, tasks are assigned such that the possibility scores for any given task are compared and optimized so that each maker can be assigned the task with the lowest possibility score possible. This ensures that both a good task for the maker, and a good maker for that task, are chosen. To avoid conflicts, the Task Assigner assumes there is at most one maker per subcomponent, i.e., multiple makers cannot work simultaneously with the same indivisible component. Since the Task Assigner does not follow a single or fixed path of fabrication, makers can proceed at whatever pace is comfortable to them, and join or drop out of the activity whenever they wish.

4.2.4 Alternative Tool Database. As the literature review on the deficiencies in static explicit knowledge resources revealed, it is common for a maker to have challenges with the tools that are specified. In some cases, this is due to the required tool being in use, being unavailable, the maker not knowing how to use it, and so on [56, 62, 82]. As part of Automatics, an Alternative Tool Database that contains a variety of common fabrication tools, and lists of alternatives to them, was created. The Task Assigner uses the database to find a new tool whenever the maker indicates that they are unable to use the suggested tool. It is also consulted when a maker has indicated that they prefer to use an alternative technique or process variant (e.g., hand tools instead of power tools) when the maker first opens a Dynamic Manual in Automatics.

4.2.5 Maker Interface. The Maker Interface is a browser-based client that the maker works with directly. The Maker Interface was implemented using CoffeeScript and JavaScript for interaction and Three.js to render the 3D artifact model. The Maker Interface interactively displays and animates tasks in a step-by-step manner, as per the fabrication path dictated by the Task Assigner (G1). The goal, tools that are required, and techniques or processes that are used for the current

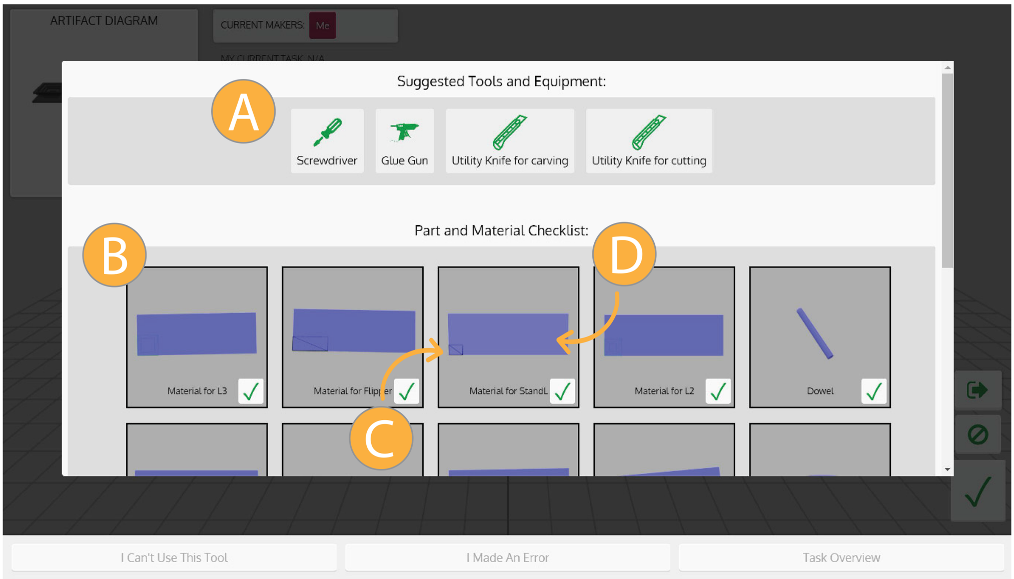


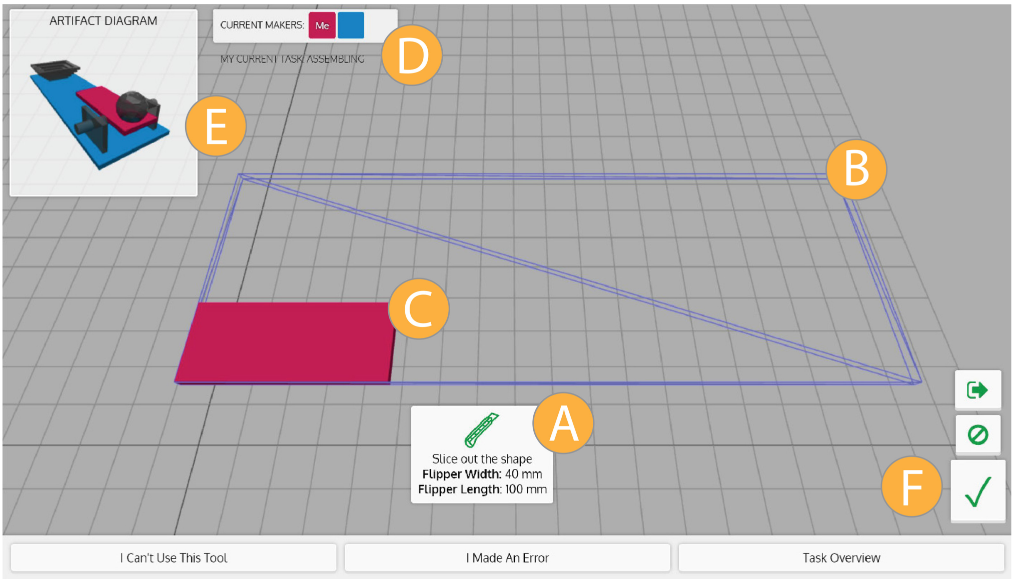
Fig. 9. The Maker Interface initially presents the maker with a list of suggested tools that could be used to complete the artifact (A), and a part and material checklist (B) that they need to complete to ensure that they have all the necessary materials and manufactured parts necessary. In the case of this artifact, the maker is shown that some pieces (C) are composed from larger raw materials (D).

task are visualized next to the current subcomponent being used or fabricated (G3). The interface also lets makers specify simple facts about their environment and process (e.g., “I can’t use this tool,” “I made an error;” G2). Based on the progress and input of makers during fabrication, infeasible pathways through the AND/OR graph are pruned. These results in the pathway that the makers are taking being dynamically determined are based on their input and choices. Once the artifact has been completed, the makers will have, thus, traversed through a single, dynamically-determined pathway from the initial raw materials and parts, to the finished artifact.

As the Maker Interface is browser-based, it can run on any platform; however, tablets were chosen as they are more ubiquitous than VR and AR systems in fabrication spaces [48]. Tablets (and laptops) are also the most popular methods for makers to access information when fabricating artifacts (e.g. downloading Instructables, troubleshooting via the internet [10, 74, 82]). Although currently running on tablets, the underlying algorithms and techniques comprising Automatics (e.g., Task Space and Task Assigner) could also be integrated within AR, VR, or smart workspaces in the future.

4.3 Maker Interface Workflow

To fabricate or assemble an artifact using Automatics, a maker selects an artifact to fabricate from the list of available Dynamic Manuals in the Maker Interface. Any additional makers who wish to join can do so at any time by selecting the same artifact as the first maker (G4). Once a dynamic manual is selected, the maker is then presented with a list of suggested tools and equipment that they may need (Figure 9(A)). The tool and equipment list is provided for informational purposes, rather than as a checklist that must be completed, because if a suggested tool is unavailable, Automatics can suggest an alternative tool that can be used. Below the list is a checklist of parts and materials that are needed to complete the artifact (Figure 9(B)). In the case of the materials



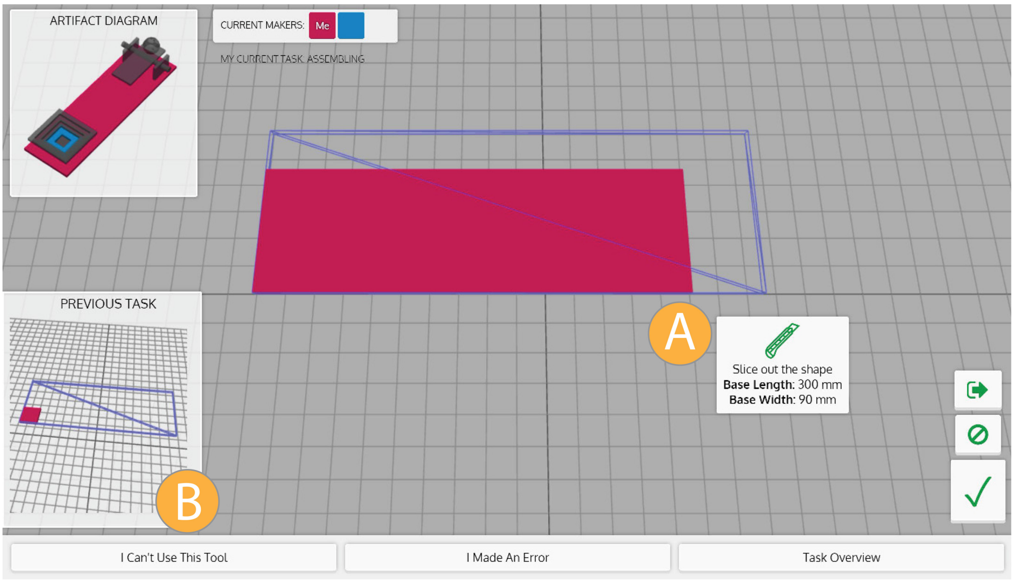
Slice out the shape
Flipper Width: 40 mm
Flipper Length: 100 mm

Fig. 10. The Maker Interface. In this example, the maker is instructed to use a utility knife to cut out a new piece for their artifact that is 40 mm wide and 100 mm long (A, C), from a larger piece of raw material (B). The interface also shows the maker how many other makers are participating (D), and illustrates the specific aspect of the artifact each is working on in the Artifact Diagram (E). To not distract the maker from their current task or draw attention away from the center of the screen, the additional options available to a maker (i.e., move on to the next task, forfeit a task, ask for an alternative tool, etc.), are provided in menus along the right side and bottom (F) of the interface.

and parts, because some parts are manufactured by other entities, and material thicknesses (among other features) are important for alignment and attachment, a checklist ensures that every subcomponent that is needed can either be made or is on hand [44]. This checklist prevents makers from encountering a task that requires them to fabricate a component that they will be unable to, thereby decreasing their frustration and the likelihood that they will terminate the process early.

Before assigning the first task, the Maker Interface then asks the maker if they want to fabricate their artifact using traditional craft methods (G2) [67]. If they opt to, Automatics will replace tasks involving power tools with an equivalent hand tool. Other process variants such as "use sustainable materials," "fastest time," "child-friendly," and the like could be added in the future.

The currently assigned task is shown in the main view of the Maker Interface (Figure 10). This view includes the subcomponent being created or used and a tool tip with necessary tool, dimensional, orientation, and operational information (Figure 10(A)). If a subcomponent is to be created from a raw material, the raw material will be shown as a wireframe model and the desired subcomponent as a solid mesh, in a visually distinguishable color (Figure 10(B, C) [44]). If necessary, an animation shows the direction the subcomponent should be attached to the existing component assembly, to ease understanding about the mode of attachment [25]. When more than one maker is participating, each is assigned a color and the main view mesh will be displayed using that color to help the maker understand how many others are participating (Figure 10(D, E); [2]). The main view is intentionally sparse and the tooltip text terse to not overload the maker with unnecessary information that could mislead or distract them from their current task or prevent the maker from being lost in verbose descriptions [66].



Slice out the shape
Base Length: 300 mm
Base Width: 90 mm

Fig. 11. The Maker Interface showing the current task to complete (A) and a temporary view, in the Past Task Preview, of the past task the maker just completed (B).

Global Context Awareness via the Artifact Diagram and Previous Task Preview. An Artifact Diagram of all the subcomponents currently being worked on by all makers is also provided so as to not distract from the task at hand, but provide environmental information if desired (G3; Figure 10(E); [2, 4, 30]). The Artifact Diagram enables the maker to obtain a global understanding of the entire artifact that is being fabricated so that they understand the goal of their current task, how it fits into the larger global goal, and the progress of others, using a consistent color-coding scheme [2]. The camera views of the main view and Artifact Diagram models are decoupled to enable the maker to freely zoom, pan, and rotate each view as desired, enabling the maker to view different sized or located subcomponents as needed, without resulting in the opposite view providing irrelevant visual information.

Once a task is complete, the maker can tap on the checkbox in the bottom right (Figure 10(F)) to indicate that they have completed their task and are ready to be assigned a new one. A pictorial representation of the task that they just completed, i.e., the Previous Task Preview, is briefly shown on the left side so that the maker can easily refer back to what they have just completed, thus improving understanding of the tasks at hand [30] (Figure 11(B)). As the current task (Figure 11(A)) may build on the past one or be entirely different, this helps a maker to understand how tasks are related, without detracting from the current task (G3). The camera view for the Previous Task Preview is decoupled to allow the maker to freely manipulate the view, if needed.

Overcoming Tool Limitations. If a maker encounters a task they cannot perform due to tool limitations, they can indicate that they cannot use the current tool using the buttons on the bottom options bar (Figure 12(A)). In some cases, the maker may not wish to use a given tool (e.g., they do not know how or like to [82]), whereas in other cases, the tool may currently be in use [62, 82] or the maker does not have access to the tool [56]. Selecting either option will result in the maker being presented with a list of possible tool alternatives that is populated from Automatics database of tool alternatives (Figure 12(B)). Once a maker has selected an alternative tool, the tooltip in the

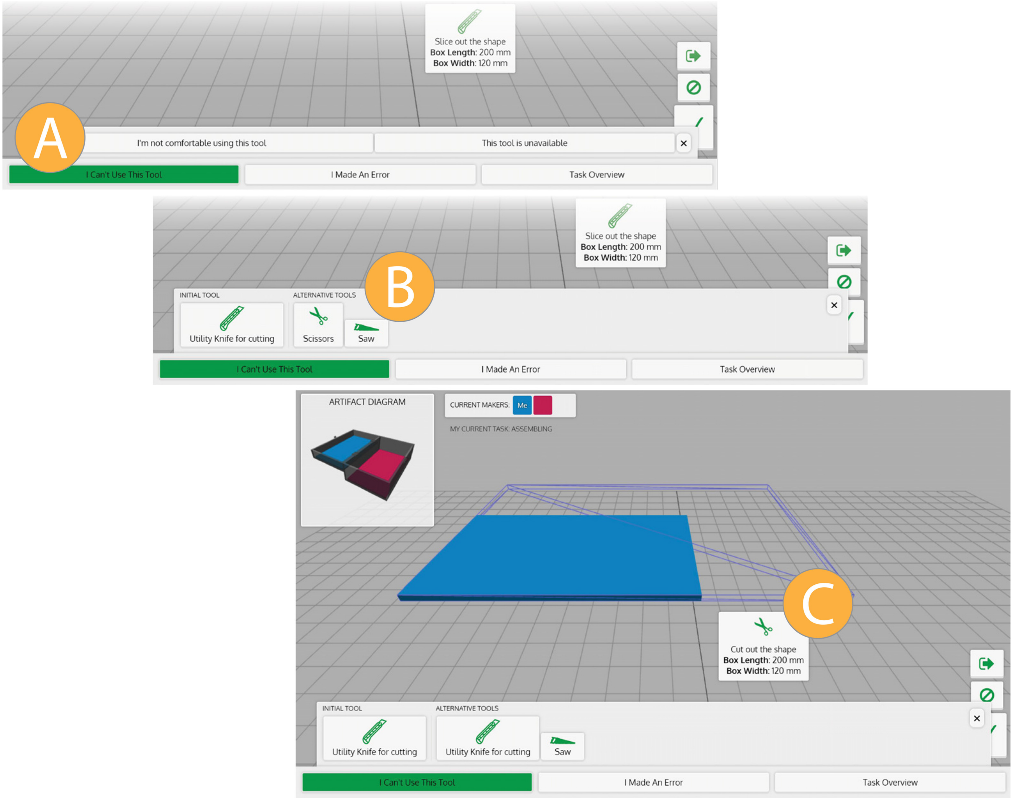


Fig. 12. When a maker is not comfortable using a tool or one is in use, they can indicate that they want a tool suggestion (A). In this example, the maker is not comfortable using the utility knife, so they are presented with a list of alternatives (B), and chose to use scissors to complete the task (C). The interface is updated, and any time another task requires the utility knife, the task will be updated to reflect their preferred tool, the scissors.

main interface will be updated to show that tool and the instructions that accompany that tool (Figure 12(C)). The newly substituted tool will continue to be used in place of the original tool in all tasks moving forward.

The exception to this occurs for tools where the maker has indicated that the tool is currently in use. In this case, the next time the tool is needed, the original, not substituted, tool will be displayed. This is done so that, when possible, the original intent of the designer is maintained. Note that Automatics does not support suggesting alternative operations, as this could change the number of tasks required to complete fabrication, and thus the Task Space graph.

Personalization and Error Recovery. During the fabrication process, it can be common for a maker to want to inject some personalization into an artifact or slightly deviate from a tutorial or sequence of tasks [50, 71, 76]. Similarly, it is also common for a maker to make a mistake during fabrication and need help [19], for example, attaching a subcomponent in error or fabricating a subcomponent to the wrong dimensions. Within the Maker Interface, deviations from the presented task can be overcome and are supported using a single process: artifact regeneration (G2). If the maker has decided that they wanted to create a subcomponent that differs in size from the one they are currently, or have previously created, they can tap on the error and personalization button on the bottom options menu (Figure 13(A)). They are then asked if the change/error occurred during the current or a past task. If it occurred on the current task, they are asked to indicate how their change differs from that in the main view using a task-specific set of options (e.g., Wrong Length? Wrong Width? Wrong Diameter?, etc.; Figure 13(B)) and enter in the new, correct dimensions that they have (Figure 13(C)). Once completed, the Maker Interface indicates that the model is being regenerated to fulfill the new requirements (i.e., the dimensions are sent back to the Automatics system and forwarded to the command-line parametric program to regenerate the model; Figure 13(D)) and will update the model and tooltip instruction once the changes have been made (Figure 13(E)).

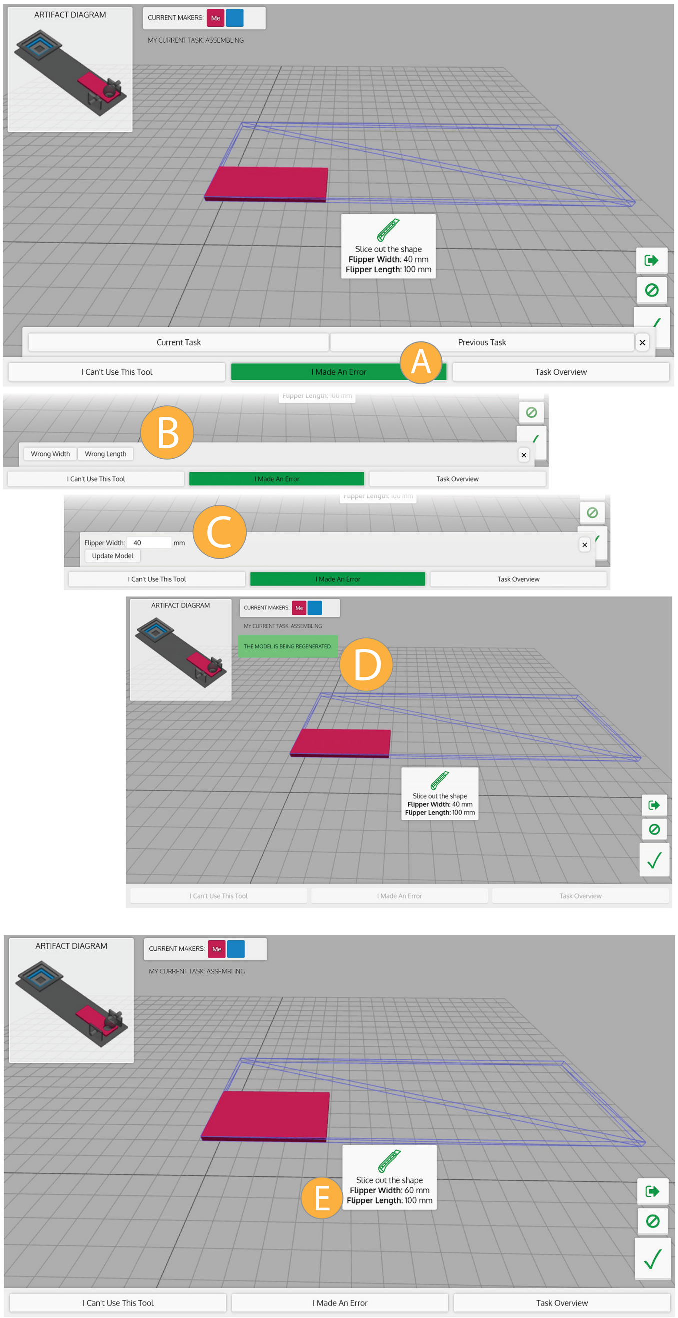
If an error was made with a subcomponent that was used in the past, the maker is then shown an overview timeline of the tasks that they have completed (similar to Figure 14) and must select the task with the desired subcomponent. Once they have corrected the error, they will then be assigned reassembly tasks, which will take them back the task that they were on when they noticed the error.

Task Overview. The Task Overview shows a maker where they are in the context of the entire fabrication process, (G3; Figure 14). It demonstrates where each maker is in the fabrication workflow, separating out each maker into their own task timeline, thereby, helping to build comprehension of the entire fabrication process [2, 4, 30]. As not all makers join the activity at the same time, the timeline may have empty cells (Figure 14(E, F)). Similarly, as not all makers work at the same pace, some tasks may appear to be repeated (Figure 14(B, C, D)), to account for another maker completing multiple tasks in the time it takes the current maker to complete one (Figure 14(G, H)). All camera views in the Task Overview are linked so that a maker can view and manipulate all tasks from the same vantage point. This enables the maker to maintain a consistent mental model of the past, current, and future tasks while comparing them throughout the visualization [24, 25]. While this may result in some tasks initially being out of view, it provides cohesion and consistency across the multiple of views the maker is presented with, and thus helps to build their understanding of the tasks that a maker may have already performed, or potentially perform in the future.

In addition to showing the previous (Figure 15(A, D)) and current tasks that have been completed (Figure 15(B, E)), the Task Overview also shows a simulation of the future tasks to be completed (Figure 15(C, F); [30]). As each task is assigned in real time depending on the number of makers, tools available, tasks left to be completed, and so on, it is not possible to display the definitive set of tasks that each maker will be assigned. Instead, the simulation provides one potential pathway to artifact completion. The simulation is the result of repeatedly assigning tasks to an arbitrary maker until they are unable to do any more tasks, and then continuing with arbitrary makers until fabrication would be complete. As the simulation is rerun every time a maker opens the Task Overview (and thus more tasks are complete), the visualization provides a real-time estimation of one possible pathway towards the successful completion of the artifact, and thus the global goal each participating maker is working towards.

In some instances, the Task Overview simulation and actual task assignment may result in a maker having to wait for others to complete tasks before they can continue (Figure 16(A, C, E)) or the maker being unable to be assigned anymore tasks (Figure 17(A–D)), as the remaining tasks should all be completed by the same maker (as per the artifact designer's original specifications for the artifact or the tasks remaining; Figure 17(E–H)). In these cases, the Task Overview visualizes these "pauses" in the process to ensure that an accurate representation of the fabrication process is maintained and presented to the maker.

Forfeiting Tasks and Finishing Fabrication. The Maker Interface also allows makers to forfeit tasks (G2), for example, if they are tired of performing the same action or using the same tool. To forfeit



MY CURRENT TASK: ASSEMBLING

Slice out the shape
Flipper Width: 40 mm
Flipper Length: 100 mm

Flipper Width: 40 mm

THE MODEL IS BEING REGENERATED.

Slice out the shape
Flipper Width: 60 mm
Flipper Length: 100 mm

Fig. 13. The Maker Interface enables a maker to fabricate even if they have made a mistake or changed their design. In this example, the maker wanted to cut a piece bigger than what the task instructed, so they (A) select that they made an error or design choice, (B) indicate that it was on the current task, and (C) enter the current width of the piece. The Maker Interface regenerates the model that is shown to the maker (D) and the task instructions are updated to reflect the new dimensions (E).

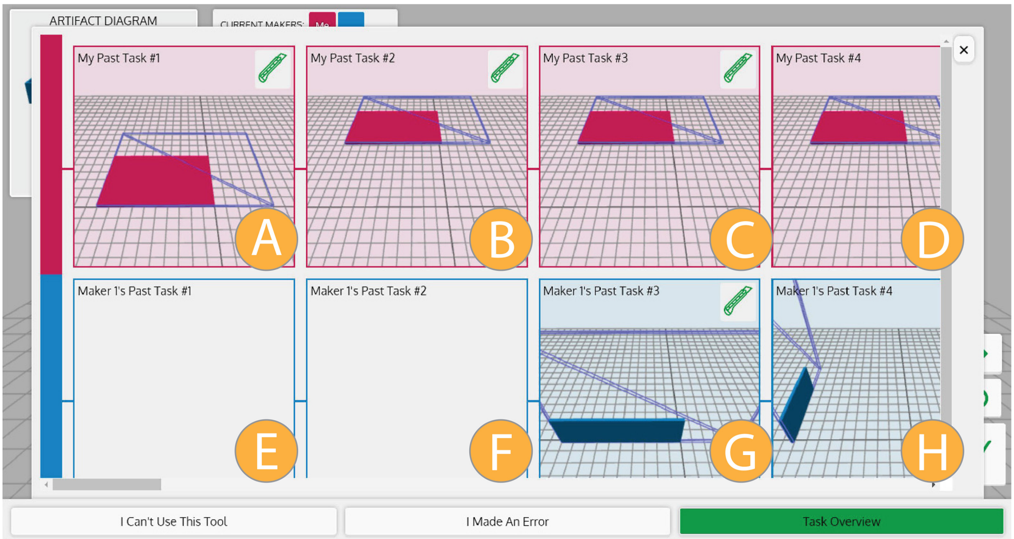


Fig. 14. (Past Task 1–4) The Task Overview shows the past, current, and simulated future tasks assigned to all makers. In this example, two makers are working together on an activity (Maker 1: A–D; Maker 2: E–H). The first maker, who is looking at the Task Overview, completed two tasks (i.e., cutting two components from raw materials in A, B) before the second maker joined (C, G). While the first maker was working on completing a task (i.e., cutting a component from a raw material in B–D), the second maker completed two tasks (i.e., cutting two components from a raw material in G–H).

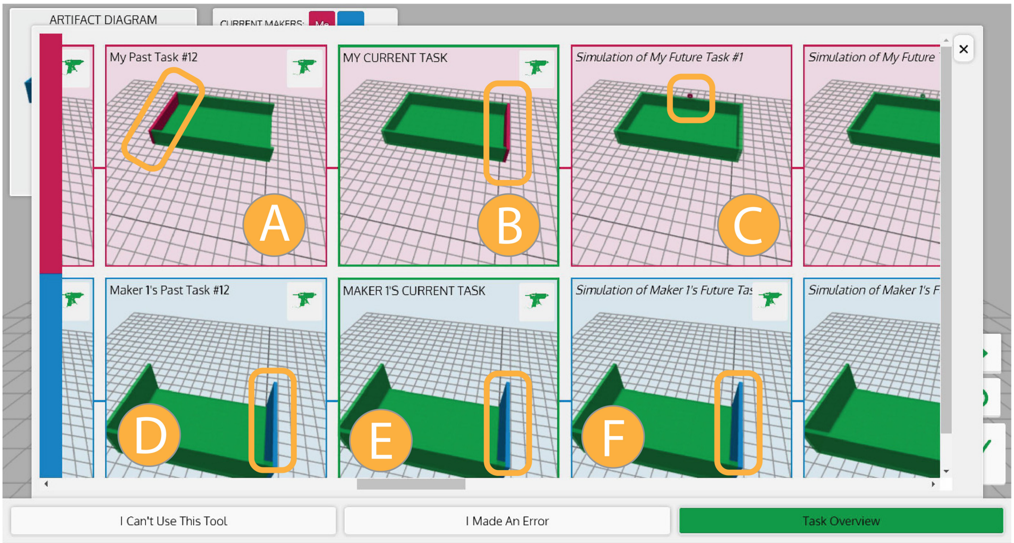


Fig. 15. (Same Context as Figure 14: Past Task 12, Current Task, Simulated Future Task 1) As the two makers continue to work on tasks (A–B, D–E), the Task Overview shows a simulation of potential tasks they could encounter if mistakes are not made, all tools are available, and no additional makers join or drop out of the activity (C, F). For clarity, the subcomponent currently being, or would be, attached in each timestep is circled in yellow.

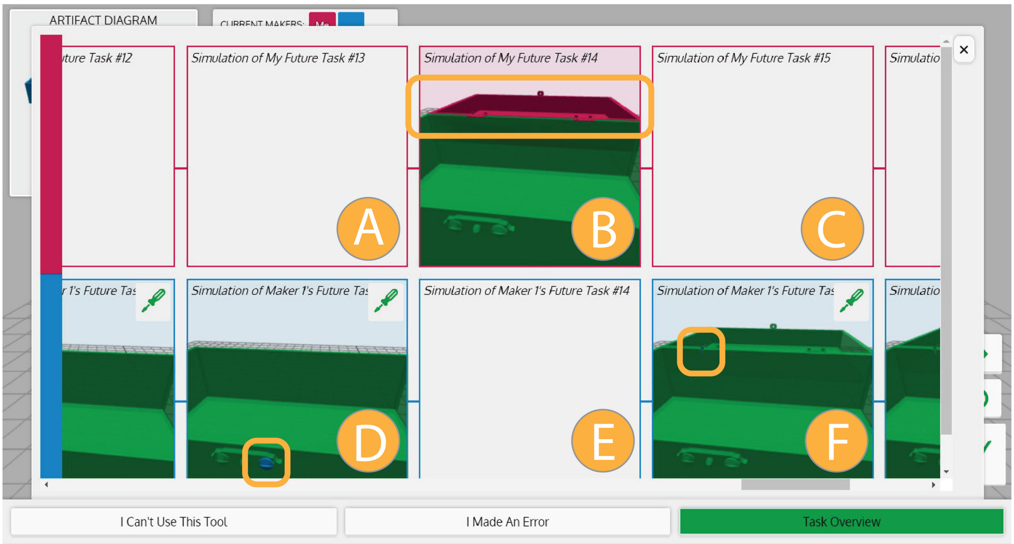


Fig. 16. (Same context as Figures 14 and 15: Simulated Future Task 13–15) As this artifact is composed of two subcomponents that must eventually come together, the simulation shows that there will be a time where the first maker (A–C) needs to wait for the second maker (D–F) to complete some tasks (i.e., A, C while waiting for D and F; E while waiting for B). For clarity, the subcomponent that would be attached in each timestep is circled in yellow.

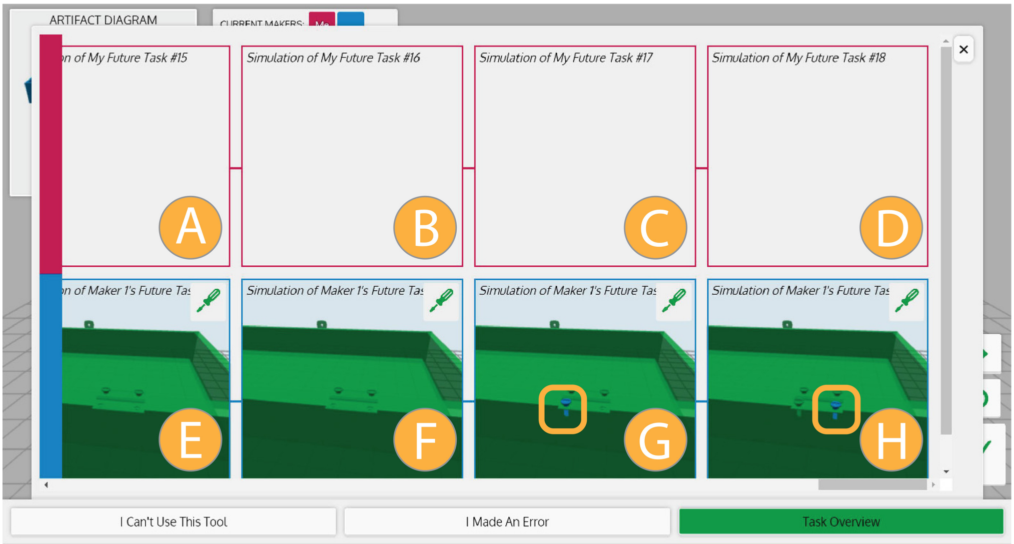


Fig. 17. (Same Context as Figures 14–16: Simulated Future Task 15–18) Towards the end of many activities there will be tasks that combine the subcomponents into a larger whole. In these cases, the first maker (A–D) will be left with no tasks to complete because they have all been assigned to another maker (E–H). This prevents both from having to share subcomponents and tools. For clarity, the subcomponent that would be attached in each timestep is circled in yellow.

a task, the maker taps on the forfeit button in the bottom right (Figure 10(F – middle button)). If a task is forfeited, a new task is assigned to the maker and the forfeited task is placed back into the pool of available tasks. If a maker has indicated that they do not want to perform a task, it will not be reassigned to the maker unless there is no other option. If a maker wishes to terminate the fabrication process early, they can exit the Maker Interface by tapping on the exit button (Figure 10(F – top button)). The task that they were assigned will be considered incomplete and put back into the pool of available tasks. As the heuristics that divide tasks between makers analyze which subcomponents attach to the subcomponent that another maker currently has, eventually, all remaining tasks will be assigned to a single maker to finish. All other participating makers will then be informed that there are no more tasks for them to complete.

5 OBSERVING AUTOMATICS' USE

Interactive assembly systems and augmented fabrication tools have already been shown in prior work to reduce assembly time and errors [4, 8, 41, 70, 87]. Rather than demonstrate that an interactive system such as Automatics outperforms static, paper-based instructions, the goal of the present evaluation was to begin to understand how dynamically generated instructions that adapt to environment and fabrication contexts could influence fabrication and making when compared to instructions that do not exhibit such dynamicism. Such an exploration has yet to be undertaken in the research literature, both when makers are making individually or collaboratively. Thus, an exploratory study was conducted to understand the differences between Automatics and traditional instruction sets that follow a prescribed sequence and do not always match the current fabrication or assembly context.

Given the diverse differences that exist between various forms of instruction presentation media, there is not an unbiased comparison that can be made. A comparison with video-based tutorials, such as YouTube tutorials, for example, would have seen variation the ability to switch between current and past tasks with a simple click, the presence of audio, and the ability of videos to convey information that 3D models and animation cannot, among others. Comparisons to how-to's or blog-style tutorials on websites such as Instructables, Adafruit, or Make.com would have additionally had challenges due to the nature of the images that are provided (e.g., photographs of the actual components versus 3D renderings of the components) and the emphasis on text-based versus illustration-based information. While paper (or digital .pdfs) may not be the most commonplace media used for making activities, it is still commonplace for assembly tasks. The comparison to paper enabled for the aforementioned issues to be controlled such that the differences between the media could be reduced to the interactive and dynamic instruction assignment differences that exist between the two. While a comparison could have been made to a tablet-based, non-interactive version of Automatics, the resulting media, while controlling for all confounds, would not have been ecologically representative of instruction media in use today.

5.1 Participants

Twelve novice makers with little fabrication or assembly experience were recruited to participate in the exploratory study (M = 28 years, range 18–51 years; 4 females; 4 assembled or fabricated things once a month, 8 once a year, or less). This population was chosen as they are the target audience of many tutorials and how-to's today, and are prone to making mistakes and misreading instructions. As the study was exploratory in nature and fabrication tasks are often performed individually or collaboratively, 12 makers were recruited. This was so three pairs of participants (i.e., three groups of two people, for a total of six) could be observed using Automatic's dynamic task assignment functions and another six people could be observed using an Automatic by themselves. Each participant was provided a $20 honorarium after their completion of the one-hour experiment.

5.2 Equipment

Although virtual or augmented reality systems for assembly have been presented in prior work, they have yet to become commonplace for novices or everyday makers. As the everyday person is familiar with, and often has only encountered paper-based instruction manuals, Instructables, or how-to videos, two explicit knowledge resources were used in the study: Automatics and paper instructions. When using Automatics, each participant was provided with an 11" × 7" tablet that responded to touch input and enabled participants to forfeit a task, provide input about the current environmental context, and indicate when they have made a mistake, among other functionality. When using paper, each task instruction was generated from a screenshot of the Maker Interface to maintain visual consistency. The Artifact Diagram and main view tooltips were visible and the main view model was zoomed in to fit within the 11" × 8.5" page. The viewpoint of the model was held consistent throughout each screenshot. Removed from the paper version were the bottom menu options (which could not have been interacted with in the paper version), the Task Overview window (which would have been static and unusable in the paper version), and the Previous Task Preview that would have shown the past task the maker just completed (which in Automatics only stayed on the screen for a few moments). Participants were provided with all the tools and raw materials needed for fabrication (e.g., glue gun and glue sticks, ruler, craft knives, screw drivers, pencils and markers, Styrofoam, foam core boards, and wooden dowelling), except for a hole punch (so that they would need to ask for an alternative tool).

5.3 Artifacts

Participants fabricated two artifacts: a small jewelry box and a ball catapult game (Figure 18). The artifacts required 10 subcomponents to be measured and cut from the foam core, and the integration of a few "additional" parts (i.e., two hinges and a lock for the jewelry box and dowelling and Styrofoam for the ball catapult). Foam core and glue were used in place of wood and nails for safety reasons.

5.4 Procedure

The experiment took place within a makerspace environment, with six participants working individually to fabricate each artifact on their own, and another six working in pairs (i.e., seated next to each other) to fabricate one instantiation of each artifact. Each participant had their own set of materials, tools, and a tablet with the Dynamic Manuals and Automatics on it. Each participant working by themselves was given their own set of paper instructions. Much like what is found today with product instructions, partnered participants were given one set of paper instructions to share. Partnered participants were encouraged to communicate and work with each other. Observing both individual and partner conditions enabled for an understanding of both Automatics' dynamic, responsive elements in the individual and paired conditions and also its task division functions in the paired condition.

Participants were given one of the explicit knowledge resources (i.e., the paper instructions or Automatics) and assigned one of the two artifacts to fabricate. They were then shown a short demo of the resource that they would be using and were given approximately 25 minutes for fabrication. Once elapsed, participants' workspaces were reset, participants were given the other explicit knowledge resource that they had not yet used (i.e., Automatics or the paper instructions), a demo of the other medium was given, and approximately 25 minutes was allotted for the fabrication

Fig. 18. The two artifacts that were fabricated during the study, the Ball Catapult Game (A) and the Jewelry Box (B).

of the second artifact. After the session, participants completed a short Likert-based survey (*1 – Strongly Disagree* to *7 – Strongly Agree*) and questions from the NASA-TLX about their experiences. Partnered participants also completed additional questions about Automatics multi-maker functions. Across all sessions, the presentation order of explicit knowledge resources was randomized, as was the order in which each artifact was fabricated.

5.5 Observations

As participants were divided into individual and partner conditions, the number of participants in each condition was too small to analyze statistically. Descriptive statistics are provided in lieu.

5.5.1 Task Completion and Mistakes. When looking at the percentage of tasks that participants completed, all participants, except for one pair (P1), completed more tasks when using Automatics (*Mdn* = 53.85%, *IQR* = 18.06) than paper (*Mdn* = 34.78%, *IQR* = 28.26; Figure 19). This can be attributed to the "*new and innovative*" way participants could explore the information that was presented for a given task. Many participants felt that Automatics required less mental effort than paper (*Automatics Mdn* = 6, *IQR* = 2.13; *Paper Mdn* = 8, *IQR* = 5.125) and were easier to use (*Mdn* = 5, *IQR* = 1.25). One participant noted that, "*With Automatics, I can accomplish the task faster. I don't have to double check if I made a mistake … it is more clear and straight forward compared to paper based … I can see [with Automatics] what I am trying to build*" (G4).

Across all sessions, participants made few mistakes and felt that Automatics decreased frustration while fabricating (*Automatics Mdn* = 7, *IQR* = 4.63; *Paper Mdn* = 9, *IQR* = 5.88). Of the mistakes that were observed, participants cut subcomponents to the wrong size, swapped the length and width dimensions, or placed glue on the wrong face of a subcomponent. Of those who made measurement mistakes, instead of regenerating the model, they decided to simply cut the piece again. Although participants were told that the model regeneration could handle a number of mistakes,

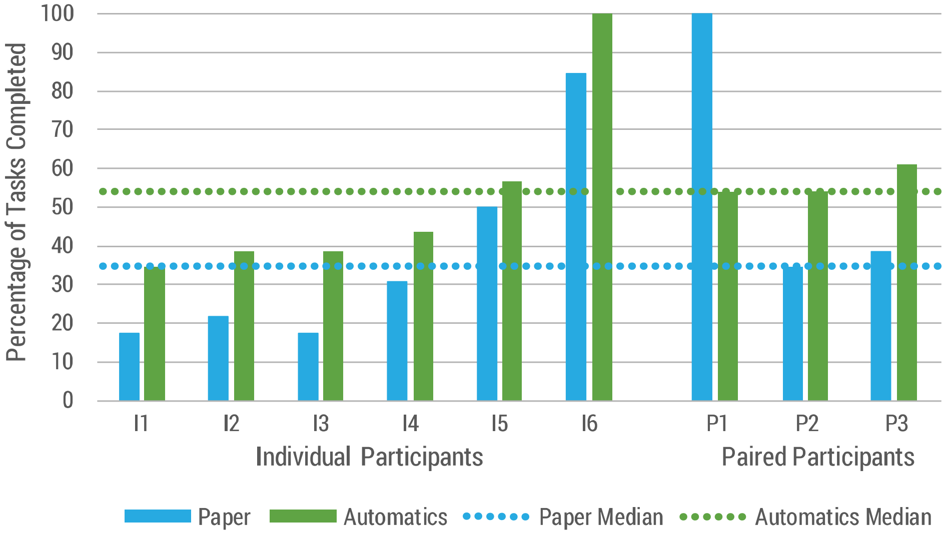


Fig. 19. The percentage of tasks that participants completed while using Paper versus Automatics in the individual (I1–I6) and paired conditions (P1–P3).

participants may have viewed model regeneration as only appropriate for serious, irreversible mistakes, rather than a feature that they could use to overcome any mistake, even one that could be fixed by quickly measuring and re-cutting a subcomponent.

Other mistakes were the result of being able to deviate from the task sequence. One participant, who made gluing mistakes, grouped each of the paper instructions by the operation to be performed (e.g., measuring, cutting, and gluing) and then completed all tasks that required the same operation. While personally satisfying, this ignored the dependencies required to fabricate and resulted in her having glue on the inside and outside of the artifact. Automatics' ability to reorder tasks and complete fabrication via sub-assemblies prevents this potentially error-prone workflow (G2), however this behavioral observation suggests that it would be useful to consider additional Task Assigner heuristics, for example, assigning tasks-based operation or material preferences.

5.5.2 Medium-Specific Behaviors. When using paper, 7 of the 12 participants fanned the pages to compare tasks or arranged the pages into a spatial explosion diagram. This allowed them to build up a contextual understanding of the current task and how each task fit within the global context of the final artifact. Such behaviors were not observed when using Automatics, likely because task comparison was supported via the Task Overview and the Artifact Diagram. Because these two elements provide a global understanding of the current task, makers don't have to scan across their workbench or flip between pages to gain understanding. This was echoed by participants, one of whom noted that they liked the Artifact Diagram and Task Overview because "*I could see the final result beside the current step so I could see exactly what each step was for*" (*Artifact Diagram Mdn* = 5, *IQR* = 3; *Task Overview Mdn* = 5, *IQR* = 1; G3). Given the spatial nature of some makers, it would, however, be beneficial to consider alternative visualization techniques that would allow tasks to be rearranged into explosion-style diagrams or layouts that would be more conducive to their understanding of how the artifact was composed.

5.5.3 Multi-Maker Workflows. Those participants who fabricated with a partner were enthusiastic about the automatic division of tasks (*Mdn* = 6, *IQR* = 1; G4); however, we did not observe a doubling or tripling in the number of tasks that could be completed individually versus with a partner [9]. The pair of participants (P1) who completed more tasks with paper than Automatics did so based on how they divided the tasks. One participant behaved like the lead maker, opting to perform most of the tasks himself, "hogging" a number of subcomponents, and delegating only a small number of tasks to his partner. Although his pair completed more tasks, animosity grew between them and was easy to observe. When this pair worked with Automatics, the automatic division of tasks enabled both participants to contribute equally and the forfeiting of tasks helped eliminate their interpersonal conflicts (e.g., "*it was good that it also gave alternative steps if I could not perform one*" (G2, G4)). With the other two groups of participants, neither pair worked such that one individual in the pair was in charge or more important than the other.

While using paper, all participants divided the paper instruction booklet in half, or into logical chunks. One group did this automatically right at the beginning of the task, whereas, another group kept the instructions together as a unit and removed pages little by little. This second group would work on a few tasks and then come back together, divide up a few more tasks, and work on the tasks individually, come back together, and so on. The third group initially kept the instructions together and started duplicating each other, but eventually divided the instructions between themselves. Once the instructions were divided, participants then had conversations about how the subcomponents contained within each chunk would need to fit together once they were all fabricated. These conversations were either at the beginning of the process, or once one participant had gotten to a point that they need a component that the other had. With Automatics, this process was performed automatically, so participants were free to begin working almost immediately, without having a long consultation and task division period.

Although the initial consultation period may have been shorter, when using Automatics, participants were observed to be more likely to consult with each other throughout the process when compared to when they were using paper. One member of the pair often picked up their tablet and leaned over to their partner to compare screens or talked about the Task Overview tasks that were being simulated. With Automatics, participants didn't know which tasks they would be assigned, so this naturally encouraged conversation and consultation. With paper, once the tasks were divided between participants, there was little conversation between participants, likely because they both knew which activities they needed to perform, so there was not a need for further consultation.

5.5.4 Reappropriating Automatics Functionality. In total, 67% of participants generalized Automatics applicability to other activities, such as baking or cooking, suggesting that they would be helpful to "*automatically assign different steps to different people who want to cook together.*" Others suggested activities "*whose instructions are notoriously difficult to follow, like furniture construction,*" conducting science experiments, or during multi-user stage setups. All participants indicated that they wanted to use Automatics in the future ($Mdn = 6$, $IQR = 2$), alluding to the benefits makers see in dynamically generated explicit knowledge resources such as Automatics.

5.6 Discussion

As discussed at the outset, the goal of the presented study was not to demonstrate that paper-based instructions are inferior to tablet-based instructions that have dynamic elements. Rather, it was to gain an initial understanding of how participants who had little to no experience making and also had little exposure to using anything more than paper-based procedural instructions would (i) utilize such resources and (ii) how these resources would influence their workflows. This is an obvious limitation of the study, as it focused on a small group of participants who could be considered casual makers at best. There is still value in the comments they provided, the behaviors they exerted, and their abilities to complete the tasks they were given, however, because it is this population that STEM-based outreach activities and many makerspaces are trying to encourage to participate in making.

Generally speaking, the results revealed that dynamically generated instructions are a welcome alternative to traditional paper-based instruction sets. Not only were makers able to complete more tasks using this new medium, they also found that the tasks they completed required less effort, were less frustrating, and seemed easier. This is a welcome finding, as makers who are able to complete more tasks and not have to exert additional effort would be more likely to have increased motivation throughout their making process, and thus enhanced feelings of self-satisfaction. For populations of makers similar to those recruited for this study, these findings suggest that an instructional medium that is able to adapt to a maker, and thus give them a more personalized fabrication experience, may be an appropriate tool by which to introduce novice makers to physical fabrication skills and assembly activities.

The results also suggested that there is a need to support more varied pathways to artifact completion than those currently implemented within the Task Assigner. The four components that currently make up the possibility score, i.e., the opt-out score, disassembly score, tool continuation score, and component similarity score, capture a variety of workflows that a designer may want to support or a maker may want to use, however, the addition of a heuristic to enable a maker to continue working with same material or perform the same operation on a group of subassemblies, would be useful to implement. Multi-maker heuristics could be worthwhile to integrate as well, perhaps to encourage makers to come together after every few tasks and, instead of dividing up more tasks (as was observed with one group of paired participants), talk about the types of tasks, materials, or tools that each individual maker may want to utilize. Automatics does not currently allow for the adaptive weighing of heuristics during a fabrication activity, but the observed behaviors suggest that this may be a welcome addition to the Task Assigner and Maker Interface.

We also note that because participating makers in the paired conditions were strangers, they did not have a pre-existing relationship. This likely influenced the social and collaborative behaviors that were observed. Due to the number of participants, generalizations cannot be regarding the degree to which Automatics does or does not encourage collaboration or specific patterns of labor throughout a fabrication task. The initial observations, however, do appear to suggest that the uncertainty surrounding which task may be assigned may encourage more conversation throughout a making process. Future work should thus explore if this holds true, as if it does, additional algorithm heuristics and visualization elements could be added into the Task Assigner and Maker Interface, respectively, to encourage or discourage such behavior.

6 LIMITATIONS AND THE FUTURE OF MAKING WORKFLOWS

Based on the process of developing Automatics and maker feedback that was gathered, explicit knowledge resources that are adaptive and responsive have shown great potential as a novel way to impart explicit knowledge for fabrication and making. They do, of course, have some drawbacks that require additional implementation and future research.

6.1 Adapting to the Maker(s)

Much like other resources, Automatics has a limitation in that it does not currently adapt to specific maker's abilities. Rather than assume each maker is identical and assign subcomponents on a per-person basis, as is done now, the Task Assigner could be modified to include heuristics that would assign tasks based on how easily tools or parts fit within a maker's hand, the amount of strength or dexterity needed to complete a task, one's preference for certain operations (e.g., painting or sanding, versus measuring, and cutting), or even how large of a workspace is available. Depending on the maker's skill level, the complexity of tasks and accompanying animations could adapt the amount of information that is shown about specific tools, parts, processes, or techniques on a per-maker basis. This could also be integrated within other Dynamic Manuals that a maker may use (e.g., make the font that is used to display the units of measurement larger if the maker has made several incorrect measurement errors in the past), or be aggregated with data from the community of makers who have also used a given Dynamic Manual. For makers who lack knowledge about a tool or process, the tooltips in the Maker Interface could also suggest external web-based videos or how-to's that provide a maker with the background knowledge they need.

The current implementation of Automatics only allows for personalization in the form of changing the size or shape of subcomponents that are being fabricated, provided that they do not require changes be made to the subcomponents that have already been assembled or created. The reason for this, is that such personalization changes or errors would require not only the underlying model to be regenerated (which is already supported), but also the derivation and integration of a new Task Space within the current Task Space. As many other forms of personalization could be desired by a maker (e.g., remixing existing artifacts and duplicating subcomponents), to overcome this limitation, extensions to the ASP techniques that were utilized will need to be explored and developed to ensure that continuity is maintained between the Task Space, Fabrication Specification file, and Parametric 3D Model every time the constraints underlying the artifact have changed, or additional tasks are dynamically added to the Task Space.

6.2 Enhancing the Design Pathways and Tools Available for Designers

Automatics can provide value to designers because many of the manual processes currently used to create instructions can be automated or eased by Automatics. The current implementation utilizes the Constraint Specifier application to gather the information that is needed by the Task Space and Task Assigner to generate each of the tasks that a maker will encounter. Rather than asking a designer to determine which task should be first and requiring them specify each subsequent task in a linear manner, the Constraint Specifier uses the 3D model that is loaded to transform the process of creating a tutorial or instruction set into essentially a series of questions that the author must specify the answers to, i.e., "Where do all the materials or parts come from?" (Material Operations Panel), "Which materials or parts need to be attached to one another?" (Attachments Panel), and "Which subcomponents should be created at the same time or assembled by the same maker?" (Subdivisions Panel). The effort to specify the constraints inherent in model is also minimal, as the Constraint Specifier reuses the 3D model the designer already created and does not require them to photograph each task in the process, draw a diagram, or write paragraphs of text to describe a task or operation. Efforts are thus not reduced to static, unchangeable output documents or other forms of media.

While the Constraint Specifier does encourage a designer to think of their fabrication task as a series of questions rather than tasks, due to the informational dependencies that exist between some of the questions (i.e., the system needs to understand which components are attached to each other before it can generate a pre-populated list of possible constraints), this cognitive shift removes the need for the designer to worry which task should be first, determining a task order, or double-checking that the task order is correct. Because the author does not need to work out an entire assembly and fabrication sequence beforehand, the author is not forced to follow a specific sequential format when specifying constraints, which in some cases, could contradict the mental model the author has about the construction of their artifact.

The limitation of this approach, however, is that not all making tutorials or assembly tasks are rooted in 3D models and geometry. The creation of an Automatic on how to throw a clay vase, for example, could not be supported in the current system unless the author made a 3D model of the vase at different points in the throwing process and modifications were made to the animations in the Maker Interface to support the continuous rotation and addition of segments of a model. The same difficulties would also be found with activities that involve flexible components such as yarn, thread, hoses, and rope. As ASP algorithms are not yet able to handle the uncertainty introduced by flexible components [17, 40], the present implementation of the Task Space, which relies on these techniques, will not be able to support such components. Additionally, because Automatics regenerates a model whenever a measurement error has been made, from a technical standpoint, models that are not parametric cannot be supported because they cannot be algorithmically regenerated as needed.

Although the present exploration into dynamic explicit knowledge resources did not specifically evaluate the usability of the Constraint Specifier, the necessity to create such a tool because one was not available, in addition to the challenges that were identified in *Section 3 Deficiencies in Explicit Knowledge Resources*, indicates that there is much room for innovation in the tools that designers can use to create instructions and tutorials. Current methods for creating a fabrication tutorial or instructions often require an author to integrate images or screenshots with text in a word processing program and print them to a .pdf that they can distribute, record and edit a video for later uploading to the web[5], use an online wizard or blog interface that enables images, videos, and external links to be embedded alongside text[6], and so on. While each of these methods have their own benefits and weaknesses (e.g., no task or continuity checking and the ability for wide distribution and sharing), they are all static resources that require the author to specify one pathway to completion and also require the author to re-record, re-photograph, or manually edit and re-upload their resource every time an update or new variation of their document is needed. Not only is this a time-consuming process, but it also prevents a maker from working through a tutorial in the manner that suits them, their skill set, or current environmental context. Thus, not only do web interfaces need to be improved to allow for multiple variants of the same tutorial to be generated, but they also need to support content that is more dynamic, adaptive, and interactive than static images and videos.

6.3 Multi-Maker Fabrication

The Task Assigner currently assumes that makers are working independently on subcomponents that will later be integrated into a unified whole by a single maker. This assumption can be limiting, however, as this is not the only manner of collaboration possible. In some situations, such as when there is a heavy subcomponent, it would be beneficial to delegate tasks such that multiple makers work together on a sub-assembly, with one holding and another fastening. Alternatively, makers may want one maker to be the master and the other an assistant or apprentice, such as when a parent and child are working together. In such scenarios, simpler tasks, or the selection and retrieval of needed subcomponents, could be delegated to the assistant.

Although the goals and motivation for participating may differ between makers, in some instances, the overall fabrication goal may be the same (e.g., build all of these shelters in the fastest time possible and use the least amount of material to save money). In such cases, the Maker Interface could query such goals at the beginning of the fabrication process, and integrate heuristics into the Task Assigner to assign tasks based on how quickly a maker has demonstrated that they could complete similar tasks in the past, or dynamically create its own sub-divisions if a designer has not already done so as part of the Fabrication Description. Aggregate data from the utilization of other Dynamic Manuals could also be useful to integrate as well.

The present implementations' reliance on a browser-based Maker Interface enables multiple collocated makers to assemble and fabricate together, however, the current implementation does

---

[5] https://www.hongkiat.com/blog/youtube-tutorial-for-beginners/.

[6] http://www.instructables.com/id/How-to-Make-an-Instructable-Using-the-New-Editor/.

not allow for remote collaborations or take into consideration the task divisions that would need to occur when makers are not collocated. For example, if the Task Generator was extended to include a "remoteness" heuristic, sub-component tasks could be delegated based on the equipment a maker has on hand, so that a maker with access to a 3D printer, for example, could complete tasks that require such equipment and later send their 3D printed subassembly to a maker who does not have one. A similar process could occur for makers who do not have the training to use specific machines or materials. Such remote fabrication could be an alternative way to remove the tool, material, or skill limitations that currently bound fabrication activities.

6.4 Integration within Future Makerspaces

A limitation of the current implementation of the Maker Interface is that a maker is required to manually input information about their environment, if they made a mistake, and so on. This requires a maker to identify if and when they made a mistake, reverse the tasks they performed to undo the mistake, and then re-perform a series of tasks to continue fabricating. With the increased sensorization of environments, which has already begun to include makerspaces and workshops [36, 52], the Maker Interface could use the data from intelligent tools and spaces to automatically detect the current state of fabrication, without needing the maker to specific details as they have to do now. For example, tracking the current location and state of tools could assist with tool wayfinding. If the monitoring systems make use of computer vision for tool and material identification, such systems could be extended to identify when mistakes have been made or provide opportunities to suggest alternative variations that could be made to the current artifact (i.e., by comparing a model of the current physical artifact with the digital one).

Although the Maker Interface requires manual input, the Dynamic Manuals and Automatics could be used as the intelligence behind other augmented in-situ, context-dependent tasks and instructions. In addition to assigning tasks, assisting with error recovery, and supporting personalization, such components could inform the visual overlays that appear near parts or tools that will be needed during a fabrication activity in the near future, or integrate with the alternative part and tool database to show real-time alternatives within a workspace if a maker is wearing an augmented reality headset. Integrating Dynamic Manuals and the Task Assigner within virtual reality systems is also possible, and could allow the author of a Dynamic Manual the opportunity to virtually walkthrough the artifact that they have modelled and specified the fabrication tasks for, before releasing the manual to the community.

6.5 Maker Motivation and Self Efficacy

One of the largest hurdles to overcome within fabrication processes and tasks is to provide sustained motivation and help makers maintain beliefs that they are skilled enough to complete the task at hand [30]. At the beginning of an activity, a maker is likely quite motivated, however, if an artifact is being duplicated many times, or if an artifact requires stressful tasks or delays, the maker will become frustrated and lose motivation [28]. Currently, the Artifact Diagram within the Maker Interface shows the entire artifact and highlights the subcomponents that are currently being worked on, rather than showing the status of one's own progress. For some this could be enough motivation to continue fabricating their artifact. For others, however, this could be considered to be a limitation, because the maker may need smaller, more manageable goals to work toward through their fabrication process. Such diagrams could, however, be modified to break the entire artifact into more digestible, smaller sub-artifacts that can be completed quicker, thus enabling the participating makers to have a continued sense of accomplishment each time that they complete a smaller sub-artifact, rather than a larger sense of satisfaction only once the entire artifact has been completed.

Extending the Task Assigner such that it can encourage task switching or interleave more complex tasks within a series of simpler ones could also be a fruitful way to improve motivation. The Maker Interface does not currently support inter-maker communication, however, adding social networking or gamification style achievements to the interface and tasks that are assigned could help to encourage artifact completion in co-located or remote multi-maker situations and thus lessen motivational limitations.

7 CONCLUSION

Inspired by the difficulties makers encounter with making and fabrication resources, a novel dynamic explicit knowledge resource, Dynamic Manuals, and prototype system, Automatics, were developed. A Dynamic Manual, which contains a parametric 3D model, a series of assembly-sequence planning-style constraints, and a Task Space of all possible fabrication pathways that could be taken to create an artifact, dynamically generates fabrication tasks and workflows that adapt to a maker's processes and environmental context.

Automatics' makes use of a Task Assigner that dynamically traverses all possible fabrication pathways in the Task Space, using heuristics and information about the environmental context to choose good tasks for makers to perform. Through Automatic's Maker Interface, the current task that a maker is assigned is visualized and can be interacted with, thus enabling makers to develop a much-needed global awareness of the past, present, and future tasks required to complete an artifact, regardless of the number of makers who are participating in the process. Dynamic Manuals and Automatics support multiple makers working together to create an artifact, the correction of mistakes, the forfeiting of tasks, the assignment of alternative tools, and the creation of a global awareness of the tasks at hand, among other features.

Initial feedback from participants who used Automatics and paper-based instructions indicated that dynamic task assignment and workflow generation were welcome concepts that decreased frustration and increased enjoyment while fabricating. Such opinions, along with the presentation of taxonomy of challenges inherent in explicit knowledge resources, should help guide the development of systems and resources that impart explicit knowledge to makers and the development of fabrication and assembly systems that are as enjoyable and error-free as possible.

REFERENCES

[1] Mario Aehnelt and Sebastian Bader. 2015. Information assistance for smart assembly stations. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, 2, 143–150.

[2] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. 2003. Designing effective step-by-step assembly instructions. *ACM Transactions on Graphics* 22, 3, 828–837. DOI: http://dx.doi.org/10.1145/882262.882352

[3] Chris Anderson. 2012. *Makers: The New Industrial Revolution*. Crown Business, New York.

[4] Stavros Antifakos, Florian Michahelles, and Bernt Schiele. 2002. Proactive instructions for furniture assembly. In *Proceedings of the International Conference on Ubiquitous Computing*. 351–360. DOI: http://dx.doi.org/10.1007/3-540-45809-3_27

[5] Kevin M. Baird and Woodrow Barfield. 1999. Evaluating the effectiveness of augmented reality displays for a manual assembly task. *Virtual Reality* 4, 4, 250–259. DOI: http://dx.doi.org/10.1007/BF01421808

[6] George R. Beiger and Marvin D. Glock. 1984/1985. The information content of picture-text instructions. *Journal of Experimental Education* 53, 68–74.

[7] Paulo Blikstein. 2013. Digital fabrication and 'making' in education: The democratization of invention. *FabLabs: Of Machines, Makers and Inventors*. Transcript Publishers, Bielefeld.

[8] A. C. Boud, David J. Haniff, Chris Baber, and S. J. Steiner. 1999. Virtual reality and augmented reality as a training tool for assembly tasks. In *Proceedings of the IEEE International Conference on Information Visualization*. 32–36. DOI: http://dx.doi.org/10.1109/IV.1999.781532

[9] Fred P. Brooks Jr. 1995. *The Mythical Man-Month* (anniversary ed.). Addison-Wesley.

[10] Bradley A. Camburn, Karen H. Sng, K. Blake Perez, Kevin Otto, Kristin L. Wood, Daniel Jensen, and Richard Crawford. 2015. The way makers prototype: Principles of DIY design. In *Proceedings of the ASME International Design Engineering*

Technical Conferences and Computers and Information in Engineering Conference, V007T06A004. DOI: http://dx.doi.org/10.1115/DETC2015-46295

[11] Timothy Campbell, Jonathan Harper, Björn Hartmann, and Eric Paulos. 2015. *Towards Digital Apprenticeship: Wearable Activity Recognition in the Workshop Setting*. Technical Report UCB/EECS-2015-172. EECS Department, University of California, Berkeley.

[12] Lee Carroll. 2004. Static versus dynamic presentation of procedural instruction: Investigating the efficacy of video-based delivery. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 48, no. 7.

[13] Justin Cheng, Jaime Teevan, Shamsi T. Iqbal, and Michael S. Bernstein. 2015. Break it down: A comparison of macro- and microtasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 4061–4064. DOI: http://dx.doi.org/10.1145/2702123.2702146

[14] Ruth C. Clark, Frank Nguyen, and John Sweller. 2006. *Efficiency in Learning: Evidence-Based Guidelines to Manage Cognitive Load*. Pfeiffer, San Francisco.

[15] Matthew A. Dalton, Audrey Desjardins, and Ron Wakkary. 2014. From DIY tutorials to DIY recipes. In *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1405–1410. DOI: http://dx.doi.org/10.1145/2559206.2581238

[16] Marie-Paule Daniel and Barbara Tversky. 2012. How to put things together. *Cognitive Processing* 13, 4, 303–319. DOI: http://dx.doi.org/10.1007/S10339-012-0521-5

[17] Carmelo Del Valle, Antonio Márquez, and Irene Barba. 2010. A CSP model for simple non-reversible and parallel repair plans. *Journal of Intelligent Manufacturing* 21, 1, 165–174. DOI: http://dx.doi.org/10.1007/S10845-008-0162-9

[18] Dingxiong Deng, Cyrus Shahabi, and Linhong Zhu. 2015. Task matching and scheduling for multiple workers in spatial crowdsourcing. In *Proceedings of the SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 21. DOI: http://dx.doi.org/10.1145/2820783.2820831

[19] Gregory Druck and Bo Pang. 2012. Spice It Up? Mining refinements to online instructions from user generated content. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Long Papers*, vol. 1. 545–553. DOI: http://dx.doi.org/10.1145/2400000.2390601

[20] Franck Ganier. 2004. Factors affecting the processing of procedural instructions: Implications for document design. *IEEE Transactions on Professional Communication* 47, 1, 15–26. DOI: http://dx.doi.org/10.1109/TPC.2004.824289

[21] Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. 2012. Duplotrack: A real-time system for authoring and guiding duplo block assembly. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. 389–402. DOI: http://dx.doi.org/10.1145/2380116.2380167

[22] Pavel Gurevich, Joel Lanir, Benjamin Cohen, and Ran Stone. 2012. TeleAdvisor: A versatile augmented reality tool for remote assistance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 619–622. DOI: http://dx.doi.org/10.1145/2207676.2207763

[23] Allie Fraser, Tovi Grossman, and George Fitzmaurice. 2017. WeBuild: Automatically distributing assembly tasks among collocated workers to improve coordination. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1817–1830. DOI: http://dx.doi.org/ 10.1145/3025453.3026036

[24] Julie Heiser, Doantam Phan, Maneesh Agrawala, Barbara Tversky, and Pat Hanrahan. 2004. Identification and validation of cognitive design principles for automated generation of assembly instructions. In *Proceedings of the Conference on Advanced Visual Interfaces*. 311–319. DOI: http://dx.doi.org/10.1145/989863.989917

[25] Julie Heiser, Barbara Tversky, Maneesh Agrawala, and Pat Hanrahan. 2003. Cognitive design principles for visualizations: Revealing and instantiating. In *Proceedings of the Cognitive Science Society Meetings*. 545–550.

[26] Steven J. Henderson and Steven K. Feiner. 2011. Augmented reality in the psychomotor phase of a procedural task. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*. 191–200. DOI: http://dx.doi.org/10.1109/ISMAR.2011.6092386

[27] Luiz S. Homem De Mello and Arthur C. Sanderson. 1990. AND/OR Graph representation of assembly plans. *IEEE Transactions on Robotics and Automation* 6, 2, 188–199. DOI: http://dx.doi.org/10.1109/70.54734

[28] Nathaniel Hudson, Celena Alcock, and Parmit K. Chilana. 2016. Understanding newcomers to 3D printing: Motivations, workflows, and barriers of casual makers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 384–396. DOI: http://dx.doi.org/10.1145/2858036.2858266

[29] Gemma K. Hugues and Victor M. González. 2009. Watch and learn: Studying the sharing of everyday knowledge with instructional videos. In *Proceedings of the Latin America Web Congress*. 138–145. DOI: http://dx.doi.org/10.1109/LA-WEB.2009.29

[30] *Instructions for Use of Products by Consumers*. IEC Standard 37:1995, 2009.

[31] *Preparation of Instructions for Use – Structuring, Content and Presentation – Part 1: General Principles and Detailed Requirements*. IEC Standard 82079-1, 2012.

[32] Sankar Jayaram, Uma Jayaram, Young Jun Kim, Charles Dechenne, Kevin W. Lyons, Craig Palmer, and Tatsuki Mitsui. 2007. Industry case studies in the use of immersive virtual assembly. *Virtual Reality Journal* 11, 4, 217–228. DOI: http://dx.doi.org/10.1007/S10055-007-0070-X.

[33] Krishnanand Kaipa, Carlos Morato, Boxuan Zhao, and Satyandra K. Gupta. 2012. Instruction generation for assembly operations performed by humans. In *Proceedings of the ASME International Design Engineering Technical Conference and Computers and Information in Engineering Conference.* 1121–1130. DOI:http://dx.doi.org/10.1115/DETC2012-71266

[34] Mary Kayler, Timothy Owens, and George Meadows. 2013. Inspiring maker culture through collaboration, persistence, and failure. In *Proceedings of Society for Information Technology & Teacher Education International Conference.* 1179–1184.

[35] Sungye Kim, Insoo Woo, Ross Maciejewski, David S. Ebert, Timothy D. Ropp, and Krystal Thomas. 2010. Evaluating the effectiveness of visualization techniques for schematic diagrams in maintenance tasks. In *Proceedings of the Symposium on Applied Perception in Graphics and Visualization.* 33–40. DOI:http://dx.doi.org/10.1145/1836248.1836254

[36] Jarrod Knibbe, Tovi Grossman, and George Fitzmaurice. 2015. Smart makerspace: An immersive instructional space for physical tasks. In *Proceedings of the International Conference on Interactive Tabletops and Surfaces.* 83–92. DOI:http://dx.doi.org/10.1145/2817721.2817741

[37] Stacey Kuznetsov and Eric Paulos. 2010. Rise of the expert amateur: DIY projects, communities, and cultures. In *Proceedings of the Nordic Conference on Human-Computer Interaction: Extending Boundaries.* 295–304. DOI:http://dx.doi.org/ 10.1145/1868914.1868950

[38] Benjamin Lafreniere, Tovi Grossman, Fraser Anderson, Justin Matejka, Heather Kerrick, Danil Nagy, Lauren Vasey, Evan Atherto, Nicholas Beirne, Marcelo Coelho, Nicholas Cote, Steven Li, Andy Nogueira, Long Nguyen, Tobias Schwinn, James Stoddart, David Thomasson, Ray Wang, Thomas White, David Benjamin, Maurice Conti, Achim Menges, and George Fitzmaurice. 2016. Crowdsourced Fabrication. In *Proceedings of the Symposium on User Interface Software and Technology.* 15–28. DOI:https://doi.org/10.1145/2984511.2984553

[39] Doo Young Lee and Dong-Hee Shin. 2012. An empirical evaluation of multi-media based learning of a procedural task. *Computers in Human Behavior* 28, 1072–1081. DOI:http://dx.doi.org/10.1016/j.chb.2012.01.014

[40] Alan C. Lin and T. C. Chang. 1993. An integrated approach to automated assembly planning for three-dimensional mechanical products. *The International Journal of Production Research* 31, 5, 1201–1227. DOI:http://dx.doi.org/10.1080/00207549308956784

[41] Alfredo Liverani, Giancarlo Amati, and Gianni Caligiana. 2004. A CAD-Augmented reality integrated environment for assembly sequence check and interactive validation *Concurrent Engineering* 12, 1, 67–77. DOI:http://dx.doi.org/10.1177/1063293.04042469

[42] Faisal Luqman and Martin Griss. 2010. Overseer: A mobile context-aware collaboration and task management system for disaster response. In *Proceedings of the International Conference on Creating, Connecting, and Collaborating through Computing.* 76–82. DOI:http://dx.doi.org/10.1109/C5.2010.10

[43] Ioanna Lykourentzou and Heinz Schmitz. 2015. An online approach to task assignment and sequencing in expert crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. In press.

[44] Cortney V. Martin and Tonya L. Smith-Jackson. 2008. Evaluation of pictorial assembly instructions for young children. *Human Factors* 50, 4, 652–662. DOI:http://dx.doi.org/10.1518/001872008X288592

[45] Cortney V. Martin and Tonya L. Smith-Jackson. 2005. Analysis of assembly instructions for children: The good, the bad, and the ugly. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 49, 18, 1729–1733. DOI:http://dx.doi.org/10.1177/154193120504901811

[46] Marcel Mayer, Barbara Odenthal, Marco Faber, Carsten Winkelholz, and Christopher M. Schlick. 2014. Cognitive engineering of automated assembly processes. *Human Factors and Ergonomics in Manufacturing & Service Industries* 24, 3, 348–368. DOI:http://dx.doi.org/10.1002/hfm.20390

[47] Peter Mohr, Bernhard Kerbl, Michael Donoser, Dieter Schmalstieg, and Denis Kalkofen. 2015. Retargeting technical documentation to augmented reality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 3337–3346. DOI:https://dx.doi.org/10.1145/2702123.2702490

[48] Heather Moorefield-Lang. 2015. Making, libraries, and literacies. *Library Media Connection* 33, 4, 30–31.

[49] Roger W. Morrell and Denise C. Park. 1993. The effects of age, illustrations, and task variables on the performance of procedural assembly tasks. *Psychology and Aging* 8, 3, 389–399.

[50] Kedar S. Naphade, Robert H. Storer, and S. David Wu. 1999. *Graph-Theoretic Generation of Assembly Plans, Part I: Correct Generation of Precedence Graphs.* Technical Report. IMSE Technical Report 99T-003, Lehigh University, Bethlehem, Pennsylvania.

[51] Laura R. Novick and Douglas L. Morse. 2000. Folding a fish, making a mushroom: The role of diagrams in executing assembly procedures. *Memory & Cognition* 28, 7, 1242–1256. DOI:http://dx.doi.org/10.3758/BF03211824

[52] Brian O'Connell. 2017. *Design and Analysis of an IoT Usage Tracking and Equipment Management System Within a University Makerspace.* Ph.D. Dissertation, Tufts University.

[53] Lora Oehlberg, Wesley Willett, and Wendy E. Mackay. 2015. Patterns of physical design remixing in online maker communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 639–648. DOI:http://dx.doi.org/10.1145/2702123.2702175

[54] S. K. Ong and Z. B. Wang. 2011. Augmented assembly technologies based on 3D bare-hand interaction. *CIRP Annals-Manufacturing Technology* 60, 1, 1–4. DOI: http://dx.doi.org/10.1016/j.cirp.2011.03.001

[55] Anna-Lisa Osvalder and Pernilla Ulfvengren. 2009. Human-technology systems. *Work and Technology on Human Terms*. 339–461. Prevent, Stockholm.

[56] Babasile D. Osunyomi, Tobias Redlich, Sonja Buxbaum-Conradi, Manuel Moritz, and Jens P. Wulfsberg. 2016. Impact of the fablab ecosystem in the sustainable value creation process. *OIDA International Journal of Sustainable Development* 9, 1, 21–36.

[57] Vaibhav Rajan, Sakyajit Bhattacharya, L. Elisa Celis, Deepthi Chander, Koustuv Dasgupta, and Saraschandra Karanam. 2013. CrowdControl: An online learning approach for optimal task scheduling in a dynamic crowd platform. In *Proceedings of the Machine Learning Meets Crowdsourcing Workshop at the International Conference on Machine Learning*.

[58] Miles Richardson, Gary Jones, and Mark Torrance. 2004. Identifying the task variables that influence perceived object assembly complexity. *Ergonomics* 47, 9, 945–964. DOI: http://doi.dx.org/ 10.1080/00140130410001686339

[59] Michael A. Rodriguez. 2002. Development of diagrammatic procedural instructions for performing complex one-time tasks. *International Journal of Human-Computer Interaction* 14, 3–4, 405–422. DOI: http://dx.doi.org/10.1080/10447318.2002.9669127

[60] Bruce Romney, Cyprien Godard, Michael Goldwasser, and G. Ramkumar. 1995. An efficient system for geometric assembly sequence generation and evaluation. In *Proceedings of the International Computers in Engineering Conference*. 699–712.

[61] Daniela K. Rosner and Morgan Ames. 2014. Designing for Repair? Infrastructures and Materialities of Breakdown. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing*. 319-331. DOI: http://dx.doi.org/10.1145/2531602.2531692

[62] Daniel Saakes. 2009. Big Lampan Lamps: Designing for DIY. In *Proceedings of the ACM Conference on Creativity and Cognition*. 403–404. DOI: http://dx.doi.org/10.1145/1640233.1640322

[63] Adam Sadilek, John Krumm, and Eric Horvitz. 2013. CrowdPhysics: Planned and Opportunistic Crowdsourcing for Physical Tasks. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 424, 125–620.

[64] Eldon Schoop, Michelle Nguyen, Daniel Lim, Valkyrie Savage, Sean Follmer, and Björn Hartmann. 2016. Drill Sergeant: Supporting physical construction projects through an ecosystem of augmented tools. In *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1607–1614. DOI: http://dx.doi.org/10.1145/2851581.2892429

[65] Abhishek Seth, Judy M. Vance, and James H. Oliver. 2011. Virtual Reality for Assembly Methods Prototyping: A Review. *Virtual Reality* 15, 1, 5–20. DOI: http://dx.doi.org/10.1007/S10055-009-0153-Y

[66] Camilla Söderberg, Anna Johansson, and Sandra Mattsson. 2014. Design of Simple Guidelines to Improve Assembly Instructions and Operator Performance. In *Proceedings of the Swedish Production Symposium*.

[67] Marc Spagnuolo. 2013. *Hybrid Woodworking: Blending Power & Hand Tools for Quick, Quality Furniture*. Popular Woodworking Books.

[68] Daniela Straub. 2010. Wenn Anleitungen Mangelhaft Sind. *Fachzeitschrift Technische Kommunikation* 32, 4, 49–55.

[69] Anselm Strauss and Juliet Corbin. 1990. *Basics of Qualitative Research*. 15. Sage, Newberry Park.

[70] John Strobel and Guy W. Zimmerman. 2011. Effectiveness of paper, VR and stereo-VR in the delivery of instructions for assembly tasks. *Journal of Computer Information Systems and Industrial Management Applications* 3, 11, 578–585.

[71] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. 2003. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 73–80. DOI: http://dx.doi.org/10.1145/642611.642626

[72] Joshua G. Tanenbaum, Amanda M. Williams, Audrey Desjardins, and Karen Tanenbaum. 2013. Democratizing technology: Pleasure, utility and expressiveness in DIY and maker practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2603–2612. DOI: http://dx.doi.org/10.1145/2470654.2481360

[73] Markus Tatzgern, Denis Kalkofen, and Dieter Schmalstieg. 2010. Compact explosion diagrams. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering*. 17–26. DOI: http://dx.doi.org/10.1145/1809939.1809942

[74] Cristen Torrey, Elizabeth F. Churchill, and David W. Mcdonald. 2009. Learning how: The search for craft knowledge on the internet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1371–1380. DOI: http://dx.doi.org/10.1145/1518701.1518908

[75] Cristen Torrey, David W. McDonald, Bill N. Schilit, and Sara Bly. 2007. How-to pages: Informal systems of expertise sharing. In *Proceedings of the European Conference on Computer-Supported Cooperative Work*. 391–410. DOI: http://dx.doi.org/10.1007/978-1-84800-031-5_21

[76] Tiffany Tseng. 2015. Making make-throughs: Supporting young makers sharing design process. In *Proceedings of Fablearn*.

[77] Tiffany Tseng, Robert Hemsley, and Mitchel Resnick. 2012. Replay: A self-documenting construction kit. In *Proceedings of the International Conference on Interaction Design and Children*. 320–322. DOI: http://dx.doi.org/10.1145/2307096.2307156

[78] Tiffany Tseng and Mitchel Resnick. 2014. Product versus process: Representing and appropriating DIY projects online. In *Proceedings of Designing Interactive Systems*. 425–428. DOI: http://dx.doi.org/10.1145/2598510.2598540

[79] Tiffany Tseng and Geoff Tsai. 2015. Process products: Capturing design iteration with digital fabrication. In *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction*. 631–636. DOI: http://dx.doi.org/10.1145/2677199.2687891

[80] Barbara Tversky, Maneesh Agrawala, Julie Heiser, Paul Lee, Pat Hanrahan, Doantam Phan, Chris Stolte, and Marie-Paule Daniel. 2006. Cognitive design principles for automated generation of visualizations. *Applied Spatial Cognition: From Research to Cognitive Technology*, 53–75.

[81] Michelle Vazquez and Marc L. Resnick. 1997. Interfaces for advanced manufacturing technology. In *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems*. 329–330. DOI: http://dx.doi.org/10.1145/1120212.1120418

[82] Ron Wakkary, Markus Lorenz Schilling, Matthew A. Dalton, Sabrina Hauser, Audrey Desjardins, Xiao Zhang, and Henry W. J. Lin. 2015. Tutorial authorship and hybrid designers: The joy (and frustration) of DIY tutorials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 609–618. DOI: http://dx.doi.org/10.1145/2702123.2702550

[83] Ron Wakkary and Karen Tanenbaum. 2009. A sustainable identity: The creativity of an everyday designer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 365–374. DOI: http://dx.doi.org/ 10.1145/1518701.1518761

[84] Tricia Wang and Joseph Kaye. 2011. Inventive leisure practices: understanding hacking communities as sites of sharing and innovation. In *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems*. 263–272. DOI: http://dx.doi.org/10.1145/1979742.1979615

[85] Xuan Wang, Patricia S. K. Ong, and Andrew Y. C. Nee. 2016. Multi-modal augmented-reality assembly guidance based on bare-hand interface. *Advanced Engineering Informatics* 30, 3, 406–421. DOI: http://dx.doi.org/10.1016/j.aei.2016.05.004

[86] Christian Weichel, Jason Alexander, Abhijit Karnik, and Hans Gellersen. 2015. SPATA: Spatio-tangible tools for fabrication-aware design. In *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction*. 189–196. DOI: http://dx.doi.org/10.1145/2677199.2680576

[87] Stefan Wiedenmaier, Olaf Oehme, Ludger Schmidt, and Holger Luczak. 2003. Augmented Reality (AR) for assembly processes design and experimental evaluation. *International Journal of Human-Computer Interaction* 16, 3, 497–514. DOI: http://dx.doi.org/10.1207/S15327590IJHC1603_7

[88] Matthias Wille, Sascha Wischniewski, Philipp M. Scholl, and Kristof Van Laerhoven. 2014. Comparing google glass with tablet-pc as guidance system for assembling tasks. In *Proceedings of the International Conference on Wearable and Implantable Body Sensory Networks Workshops*. 38–41. DOI: http://dx.doi.org/10.1109/BSN.Workshops.2014.11

[89] Mayu M. Yamashita, Junichi Yamaoka, and Yasuaki Kakehi. 2013. Enchanted scissors: A scissor interface for support in cutting and interactive fabrication. In *Proceedings of ACM SIGGRAPH Posters*, 33. DOI: http://dx.doi.org/10.1145/2503385.2503422

[90] Nong Ye, Prashant Banerjee, Amarnath Banerjee, and Fred Dech. 1999. A comparative study of assembly planning in traditional and virtual environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 29, 4, 546–555. DOI: http://dx.doi.org/10.1109/5326.798768

[91] Hironori Yoshida, Takeo Igarashi, Yusuke Obuchi, Yosuke Takami, Jun Sato, Mika Araki, Masaaki Miki, Kosuke Nagata, Kazuhide Sakai, and Syunsuke Igarashi. 2015. Architecture-scale human-assisted additive manufacturing. *Transactions on Graphics* 34, 4, 88. DOI: http://dx.doi.org/10.1145/2766951

[92] M. L. Yuan, S. K. Ong, and Andrew Y. C. Nee. 2005. Assembly guidance in augmented reality environments using a virtual interactive tool. In *Proceedings of Innovation in Manufacturing Systems and Technology*.

[93] Guy Zimmerman, Julie Barnes, and Laura Leventhal. 2003. A comparison of the usability and effectiveness of web-based delivery of instructions for inherently-3D construction tasks on handheld and desktop computers. In *Proceedings of the International Conference on 3D Web Technology*. 49–54.

[94] Amit Zoran and Joseph A. Paradiso. 2013. FreeD: A freehand digital sculpting tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2613–2616. DOI: http://dx.doi.org/ 10.1145/2470654.2481361

Received August 2017; revised January 2018; accepted February 2018