Eugenie: Multi-Touch and Tangible Interaction for Bio-Design

Casey Grote[1], Evan Segreto[1], Johanna Okerlund[1], Robert Kincaid[2], Orit Shaer[1]

[1]Wellesley College
Wellesley, MA, USA
caseygrote@gmail.com, esegreto@wellesley.edu,
jokerlun@wellesley.edu, oshaer@wellesley.edu

[2]Agilent Technologies
Santa Clara, CA
robert_kincaid@agilent.com

ABSTRACT

We present a case study of applying TEI research to a data-intense scientific workflow that requires the exploration of large datasets through the construction of complex queries. We describe our two-year-long effort and design iterations of Eugenie, an interface for helping synthetic biologists through the collaborative and intricate process of bio-design. We introduce new interaction techniques for browsing large data sets and for constructing complex queries with active tangible tokens and an interactive tabletop. We also discuss challenges and opportunities for applying TEI to support data-driven inquiry.

Author Keywords

Physical tokens; tabletop; queries; multi-display environments; interactive surfaces.

ACM Classification Keywords

H.5.m. Information interfaces and presentation: User Interfaces---input devices and strategies, interaction styles.

General Terms

Human Factors; Design;

INTRODUCTION

Tangible and multi-touch user interfaces offer unique opportunities for enhancing collaborative learning and discovery. Several studies indicate that horizontal interfaces support active reading [12], foster collaboration [18, 22] and facilitate external cognition [14, 18].

However, while tangible and tabletop interaction have been applied to a broad range of application domains, relatively little research has been devoted to investigating these interaction styles in the context of scientific exploration [17]. Tangible and tabletop interfaces designed for supporting scientific inquiry mostly focus on the representation and manipulation of information with inherent physical or spatial structure (e.g. proteins and molecules). Our focus is on investigating the application of tangible and multi-touch interaction to data-driven inquiry, where large and abstract data sets are accessed and manipulated. Shaer et al. proposed that applying TEI research to fields and processes involving Big Data such as genomics presents an opportunity to drive forward the theory and practice of TEI [17]. They further recalled that participants in the inaugural TEI conference panel suggested that, in the future, "more complex computation should be occurring behind the tangible interface, instead of only one-to-one input-output."

In this paper, we apply TEI research to the data-intense area of Synthetic Biology. We reflect on the design, implementation, and evaluation of Eugenie, an interface for bio-design. We describe our two-year-long effort and design iterations from a visual multi-touch interface to a tangible user interface with active tokens. This paper presents four main contributions: (1) Lessons from the iterative design, implementation, and evaluation of a multi-touch and tangible interface for querying large data sets; (2) New tangible interaction techniques for navigating large hierarchical data sets using *active* tokens; (3) Tangible interaction language for forming complex queries; and (4) A discussion of challenges and opportunities for applying TEI approaches to the exploration of large data sets. We begin with a discussion of related work.

RELATED WORK

Multi-touch Interfaces for Big Data

Several multi-touch and tangible interfaces have been created to support collaborative exploration in data intense areas. For example, Block et al. created the DeepTree exhibit for science museums [4], a multi-touch tabletop interface that allows users to explore an interactive visualization of the Tree of Life. These multi-touch interfaces support collaborative and playful exploration but limit exploration through designer-defined guided discovery scenarios.

G-nome Surfer [18] and GreenTouch [22] are tabletop multi-touch applications for collaborative exploration of genomic and phenology databases. Both applications support college-level inquiry-based learning and allow open-ended data exploration. However, in both, users cannot define and set query operators directly. Morris et al. [13] surveyed the design space of collaborative tabletop search applications, reporting that most search input techniques have relied on touch, keyboard, and mouse

TEI 2015, January 15-19, 2015, Stanford, CA, USA

input. WIMP-style controls in data-intense applications have limitations including occlusion and challenges for accurate touch due to finger size [9, 24]. Recent studies have explored novel multi-touch interaction techniques that provide advantage over WIMP-style touch controls [6]. However, these techniques often suffer from low discoverability and lack of persistence [6].

Tangible Queries

Several tangible user interfaces (TUIs) have explored the use of tokens for query formulation. Navigational Blocks [5] is an early TUI for querying a database where each block represents one query parameter, its six faces representing the possible values. DataTiles [15] combines tiles – graspable windows for digital information – on top of a horizontal display. DataTiles provides a highly expressive physical interaction language but is constrained to a horizontal surface. Ullmer et al. [21] introduced two TUIs for query formulation (TQI) that use tokens to represent query parameters. One employs "parameter wheels" for fixed query parameters: the other, "parameter bars" that can be dynamically assigned to various parameters. In both interfaces, tokens are manipulated and interpreted on a series of physical constraints. Such interaction, while highly expressive, limits the portability of tokens and possibilities for collaboration. Facet-Streams [10] combines passive tokens and an interactive surface as query parameters. Stackables [11] explores the use of active tokens arranged vertically for expressing a query. We designed a system that supports complex queries and utilizes tangible interaction with *active tokens*—programmable physical objects with integrated display, sensing, or actuation technologies [23]—which are manipulated in-air and on-bezel.

Tangible Interaction with Active Tokens

Tangible Video Editor [26] employs active tokens to represent video clips. To guide users' interaction, video clip tokens are embedded in cases that resemble jigsaw puzzle pieces. SynFlo [25] is an interactive installation for introducing Synthetic Biology concepts to non-scientists. It utilizes Sifteo cubes, microcomputers that implement various gestures including shaking, flipping, tilting, neighboring, and touching [1]. The application simulates a wet-lab bench and evokes gestures such as pouring and shaking. Valdes et al. studied user expectations of a hybrid tangible and gestural language for querying large datasets [23]. They presented a vocabulary of user-defined gestures for active tokens and a characterization of the design space. Our work draws on this study and on the previous efforts we described above.

APPLICATION DOMAIN AND DESIGN GOALS

We designed Eugenie to help synthetic biologists in the intricate and data-driven workflow of bio-design. Next, we briefly explain the application domain and the bio-design process, as well as our design strategy.

Synthetic Biology

Synthetic biology is an emerging research area that couples engineering and biology with the goal of designing organisms with new specified behaviors that are useful in particular applications, including therapeutics, environmental decontamination, and *in vivo* sensing.

The field applies engineering principles such as abstraction and modularity into biological research. Synthetic biologists often solve problems by applying a forward engineering approach: composing a specification of the behavior to be designed into an organism, and then selecting genetic elements and their regulatory architecture to achieve the functional goal. Genetic elements are treated as standardized biological parts and used like "Lego Bricks" based on data about their characterized behavior.

Design Strategy

Considering the complexity of the synthetic biology domain, our design strategy combined rigorous user-centered and participatory design methods. We established a design partnership with domain scientists from a synthetic biology lab at <*Anonymous*> University, as well as with two student teams enrolled in a synthetic biology research competition (i.e. iGEM). We also collaborated closely with the developers of the Eugene [3] synthetic biology programming language.

Over the last two years, we met with our partners regularly. We conducted three workshops in which we teamed with iGEM students, received training in the Eugene programming language, and worked together to specify a wide array of biological designs. In addition, we joined our partners' research meetings and observed them throughout the bio-design process.

Bio-design

Bio-design is a central workflow in synthetic biology. It begins with a set of functional specifications to be assembled from a set of available genetic parts, and it ends with a set of exact DNA sequences to be stitched together in the laboratory. This is a collaborative and iterative workflow that is used in most synthetic biology labs and involves investigators and both graduate and undergraduate students. Because synthetic biologists design complex systems based on uncertain biological mechanisms, the bio-design process requires large design spaces to be sampled combinatorially while applying voluminous experimental design for each design candidate. Biologists define increasingly specific design parameters with each iteration, narrowing the design space. This process typically consists of three stages: (1) Research - users search for information about the structure and function of existing biological constructs; (2) Specification - users specify the desired functionality of their biological construct (e.g. a XOR gate) and draw a generic construct using visual SBOL (Synthetic Biology Open Language). They then constrain its structure and behavior using a set of rules (e.g. a biological construct that is a transcriptional unit must begin with a promoter and end with a terminator). Many synthetic biologists use Eugene [3], a domain-specific programming language, for constructing a set of rules that narrows the solution space; (3) Exploration - users explore concrete instantiations of their specification, sampling a large data set of permutations. Each construct consists of a set of available biological parts and satisfies the specified rules. Users then iterate on this paradigm, adding more rules until they reach desired results. These activities are often collaborative and typically take place in a conference room or an office space.

Interviews with biologists revealed that, despite the value of using Eugene for formally specifying design rules, many biologists—particularly novice researchers—are intimidated by the notion of writing code and thus reluctant to program in Eugene. We also found that the planning, execution, and tracking of experimental data and results are currently implemented using ad-hoc processes that limit the scale and complexity of biological design.

Design Goals

In collaboration with our design partners, we decided to focus on supporting novice scientists - iGEM participants and college students. We identified the following goals:

G 1. Facilitating an integrated and fluid workflow - supporting users' progress from the information currently available for them (e.g., a database of standardized biological parts) to the information they need (a concrete set of complex biological constructs) through a series of stages while facilitating a flexible iterative workflow that integrates disparate data sets.

G 2. Leveraging rule-based design with Eugene – allowing novices to leverage the power of rule-based bio-design with the Eugene programming language. Lowering the threshold for specifying complex biological constructs.

G 3. Fostering collaboration - the collaborative nature of bio-design implies that multiple stakeholders must participate in this process and agree on desired results that can be produced in the lab. Thus, a tool for bio-design needs to support co-located collaboration.

G 4. Facilitating constructivist learning - our goal is to support constructivist learning at the college level by providing an engaging environment that enables users to interact with the data through sensory and motor processes important for scientific thinking [2,7].

FIRST PROTOTYPE: MULTI-TOUCH INTERFACE

Design

Informed by these goals, we designed and implemented Eugenie - a collaborative, multi-touch interface for rule-based bio-design with the Eugene programming language. Our choice to design and implement Eugenie using a large *vertical* multi-touch surface was informed by current work practices of our users, who typically collaborate through side-by-side work on the whiteboard or a shared screen.

In order to harness the power of rule-based design with Eugene while eliminating the need to write Eugene code, we provided visual representations of 3000 biological parts that could be combined to represent various Eugene rules. The visual representations draw on the standard SBOL visual notation and were designed in close collaboration with the developers of the Eugene language. We conducted a series of pilot studies with users to assess and improve the expressive power and learnability of these representations.

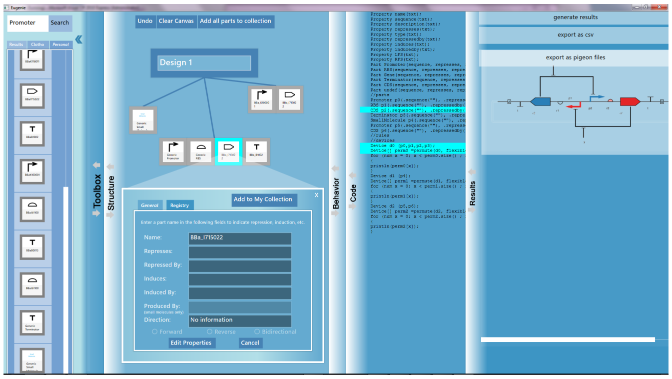
The Eugenie multi-touch application consists of a *toolbox* and a *workspace*. The *toolbox* allows users to search public or private databases of biological constructs; users can choose generic (e.g. a promoter, coding sequence, ribosomal binding site, signal, terminator) or specific biological parts and either store them in the *toolbox* or drag a copy to the workspace.

The workspace is divided into four views: *structure*, *behavior*, *code*, and *results*. Each view is represented using a sliding panel. Users can move back and forth between the different views in a nonlinear fashion, keeping one or multiple panels open at any given time, and visiting and revisiting different stages of the design process (G1). The *structure* view allows users to specify structural rules using visual representations. For example, Figure 1 shows the Eugenie system and the specification of a transcriptional unit: it must start with a promoter, contain a ribosome-binding site and at least one coding sequence, and end with a terminator. The *behavior* view enables users to specify rules regarding the interaction of two parts or constructs (e.g. signal A induces promoter A). Each representation of a biological part is associated with a *property sheet* for specifying and viewing properties and additional information. Property sheets are displayed and edited upon request, thereby reducing on-screen clutter.

The *code* view presents Eugene code that is generated automatically when users specify rules in the *structure* and *behavior* views (G2, G4). When the user selects a part in the *structure* or *behavior* view, the relevant Eugene statements in the code are highlighted. Finally, the *results* view displays the query results—all valid permutations of biological constructs that satisfy the specified rules. Permutations are displayed using standard SBOL notation.
Eugenie provides multiple points of entry by supporting parallel work across different panels (G3). The use of multi-touch input enables spatial and direct manipulation of biological constructs, which engages the connection between the hand, eye, and brain to support learning and facilitate "thinking through action" [2, 7] (G4).

Implementation

Eugenie is implemented on the Microsoft PixelSense device using MS SDK 2.0 written in C#. Information is drawn from the MIT Registry of Biological Parts, PubMed, Google Scholar, and the iGEM archive. We use the synthetic biology domain-specific languages Eugene [3] and SBOL for validating new designs (G2).



Enter a part name in the following fields to indicate repression, induction, etc.

Name: BBa_I715022
Represses:
Repressed By:
Induces:
Induced By:
Produced By: (small molecules only)
Direction: No information
○ Forward ○ Reverse ○ Bidirectional
Edit Properties
Cancel

```
Property name(txt);
Property sequence(txt);
Property description(txt);
Property represses(txt);
Property type(txt);
Property repressedby(txt);
Property induces(txt);
Property inducedby(txt);
Property LFS(txt);
Property RFS(txt);
Part Promoter(sequence, represses,
Part RBS(sequence, represses, repre
Part Gene(sequence, represses, repr
Part Terminator(sequence, represses
Part CDS(sequence, represses, repre
Part undef(sequence, represses, rep
//parts
Promoter p0(.sequence(""), .repress
RBS p1(.sequence(""), .repressedby(
CDS p2(.sequence(""), .repressedby(
Terminator p3(.sequence(""), .repre
SmallMolecule p4(.sequence(""), .re
Promoter p5(.sequence(""), .repress
CDS p6(.sequence(""), .repressedby(
//rules
//devices
Device d0 (p0,p1,p2,p3);
Device[] perm0 =permute(d0, flexibl
for (num x = 0; x < perm0.size() ;
{
println(perm0[x]);
}
Device d1 (p4);
Device[] perm1 =permute(d1, flexibl
for (num x = 0; x < perm1.size() ;
{
println(perm1[x]);
}
Device d2 (p5,p6);
Device[] perm2 =permute(d2, flexibl
for (num x = 0; x < perm2.size() ;
{
println(perm2[x]);
}
```

Figure 1, The Eugenie multi-touch interface displays *toolbox*, *structure, code,* and *results* views.

Evaluation

We conducted an observational study with 15 undergraduate science students (12 female, age 18-23). 9 students were iGEM team members, while the other 6 were majoring in life sciences. Our study focused on the *usability* of Eugenie, aiming to identify its strengths and weaknesses.

Studies were conducted in conference rooms next to users' laboratory space. In the session, we handed users written tutorial materials about bio-design, SBOL, and the Eugene programming language, which included brief paper-based practice tasks. Then, we provided an overview of the Eugenie interface, demonstrating how to specify the design of a biological construct that is functionally equivalent to an AND gate. The experimental task consisted of using the Eugenie interface to specify the design of a biological toggle switch and to explore the solution space. The task required students to: 1) search for biological parts and store them in the *toolbox*; 2) define structure rules; 3) specify relationships between parts; and 4) review results.

This task was selected, because it mirrors a real-world research task often conducted by iGEM teams. Users documented their progress and answered task-related questions. We collected data through observations and videotaped all sessions. Following the session, we debriefed the users. 12 of the users were assigned to work in dyads (total 6 dyads), while 3 iGEM students with significant Eugene programming experience worked independently.

Results

All users were able to complete the task successfully, producing biologically correct and valid designs (in some cases, with some difficulty that was resolved through collaborative work). On average, dyads spent 42 minutes working on the task, while independent users spent 41 minutes. Users' answers to a post-task questionnaire indicate that they understood the bio-design process.

Dyads collaborated effectively, mostly through side-by-side collaboration, with both users involved (physically or verbally) in the task. We observed that users used the sliding panels often and were able to easily transition back and forth between different views (G1). Users were able to specify structure rules using Eugenie without difficulties. As one user noted: "It is easier to count and relate the different parts when they are symbols rather than words. It is also easier to tell the overall structure of the system when you can see the whole thing all at once." (G4) Several users commented on the value of the *code* view. E.g.: "Seeing the code on the side and how the names were changing when you edited the properties was very helpful." (G2)

The study also highlighted several problems that led to major design iteration: (1) *Target size* – Our design incorporated WIMP-style elements such as checkboxes and textboxes within the *toolbox* and property sheets. Due to the amount of information presented, some of the controls were small; as a result, users reported difficulty in specifying properties using touch. Also, while the representations of biological parts were big enough for conveniently dragging using touch, they were still occluded by fingers. We observed that users often asked each other "which part are you talking about?"; (2) *Integrated representations* – To reduce clutter and improve the readability of visual representations, we separated the specification of structural and behavioral rules into two views. However, users noted that this separation diminished their understanding of the biological constructs designed. In addition, some users found it confusing to drag a copy of a biological part from the *toolbox* to the *structure* or *behavior* view; (3) *Persistent representation* – Our design incorporated the use of property sheets for specifying properties of biological parts. Property sheets could be displayed and edited upon request. Many users commented that some properties, such as the direction of a biological part (i.e. reverse or forward) and its size, are very important and should be represented in a persistent manner rather than hidden in a property sheet; (4) *Lack of constraints* – To allow for maximal flexibility, the representations offered by Eugenie did not utilize visual constraints. Instead, when biological parts were arranged to represent valid rules, the representations "snapped" together. Most users found it easy to specify structural rules. However, users found the specification of behavioral rules, which describe the interactions between biological parts (e.g. represses or induces) to be confusing. We think that the lack of user-guiding visual constraints in specifying the more abstract behavioral rules led to confusion regarding possibilities for action; and (5) *Aggregation* – Users expressed need for the ability to specify rules between constructs, rather than only between primitive biological parts. This requires aggregating multiple parts and rules and representing them as a single construct.

TOWARDS A TANGIBLE LANGUAGE FOR BIO-DESIGN

Design

While we could attempt to address the shortcomings described above by redesigning the existing interface, we decided to investigate an alternative approach: using tangible interaction with active tokens for constructing rules, while continuing to display Eugene code and results in a similar application to the first design on the multi-touch surface. Active tokens are programmable physical objects with integrated display, sensing, or actuation technologies that allow users to dynamically modify the tokens' associations with datasets or controls [23, 26]. We chose a tangible implementation because (1) by combining tangible active tokens with physical constraints, we attempt to provide users with persistent and integrated representations of data and with physical constraints to enforce interaction syntax (G1, G2), (2) exploring the use of active tokens on-bezel rather than on-surface allows us to design a multi-touch interface that is less cluttered and displays only essential information, and (3) tangible interaction also allows for multiple points of entry [16] (G3) as well as for direct interaction with representations through sensory and motor processes, which are important for learning [2, 7] (G4).

Our work expands on Tangible Query Interfaces (TQI) [21] and draws upon the Tangible Video Editor [26], which uses embedded active tokens in jigsaw-puzzle-inspired casings. Our design decisions were further informed by Valdes' et al. investigation of the design space of tangible interaction with active tokens combined with interactive surfaces [23]. For example, their findings indicate that users have strong preference for interacting with tangible active tokens atop a *horizontal* surface, leading us to reorient our interface horizontally. We also apply users' proposal to explore interaction *on-bezel* and *in-air* in order to free up important screen real estate. While our design implements elements that resemble tangible programming and was inspired by block-based languages, our focus is not on creating a single program; rather, we seek to allow the specification of a set of rules to query a large design space. We also incorporate dynamic bindings between active tokens and query parameters, which allows tokens to represent concrete or abstract biological parts. In the following section, we describe the interaction techniques that we developed.

Combining Active Tokens with Multi-touch Tabletop

The redesigned Eugenie application consists of a horizontal tabletop multi-touch application and a set of tangible tokens. We refer to this prototype as Eugenie++. Browsing the library of biological parts and specifying bio-design rules in Eugenie++ are performed on the tabletop bezel by creating physical configurations of tangible tokens. This allowed us to redesign the multi-touch application to include only two views: *rules* and *results*.

Eugenie++ provides two types of tangible tokens. *Active tokens*, implemented using Sifteo Cubes, represent generic or concrete biological parts that are dynamically associated by the user. The Sifteo Cubes' form factor is consistent with the synthetic biology metaphor of biological parts as building blocks. We encased each block in a cover that resembles a jigsaw puzzle piece, similar to those of the TVE [26]. The covers have connectors on the left and right, allowing pieces to interlock. Each case also has a socket on the top of the block, which is used to add operators. The active tokens can be manipulated through gestures (shake, flip), touch (swipe, click) and spatial interaction (neighbor, stack). *Passive tokens,* on the other hand, are statically bound. They represent various query operators including AND, OR, <, >, =, THEN, ALL, BEFORE, AFTER, NOT, NUMBER. These operators have sockets and plugs that indicate how they should (or should not) connect to the blocks. For example, an AND block (a binary operator) goes between two parts, while a NOT block (an unary operator) fits into the socket on top of a part. The passive tokens are shaped as symbols to indicate which operator they represent. We also use color-coding to indicate function (e.g. red for NOT).

Specifying Bio-Design Rules with Tangibles

Users specify *structure* rules by connecting active and passive tokens (G2). Users then "stamp" the physical structure within the *rules* view on the tabletop to embody the transferal of data from the pieces in the user's hand to the surface. Upon stamping, the physical representation of the rule is transformed into a digital representation in the form of a Eugene statement, which appears on the surface in the rules area. The rule is then applied to the design of the biological construct and holds unless the statement is removed from the code view. The REVERSE property is indicated by the orientation of the block. To specify behavioral rules (i.e. REPRESSES and INDUCES), users connect two parts, place them on the surface, and rotate the assembly counterclockwise or clockwise. We chose to use gestures to specify these rules, since they represent a continuous action.

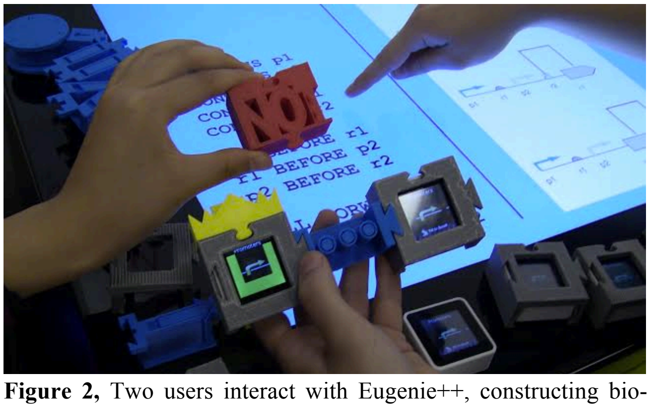


Figure 2, Two users interact with Eugenie++, constructing bio-design rules using tangible tokens. Upon "stamping" a rule on the surface, a new Eugene statement is added to the *rules* panel.

Searching Hierarchical Database with Active Tokens

Active tokens can be used to search a hierarchical database: the MIT Registry of Biological Parts, which contains over 3000 parts (G1). To browse between items at the same level of the database hierarchy, users tilt the block, like manipulating a stream of menu items. Users then neighbor the tokens vertically to traverse the hierarchy. For example, when a token is neighbored below another token that displays the generic biological part "promoter", sub-categories that exist within the class "promoter" are displayed (see Figure 3). This interaction mirrors the tree-like structure of the database as the user traverses the levels.

Aggregating Representations

Users may collapse or expand biological parts displayed by different tokens into one biological construct by stacking: simpler constructs stacked atop more complex constructs collapses, while the opposite expands (see Figure 3).

Exploring the solution space

Solutions—permutations of biological constructs that satisfy the set of Eugene rules—are displayed on the surface using SBOL notation and can be manipulated through touch in the tabletop application.

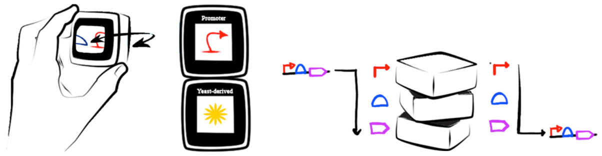


Figure 3, Illustrates tilting to search, neighboring for traversing the hierarchical database, and stacking to collapse and expand biological constructs.

Implementation

Eugenie++ uses Sifteo cubes ver. 2.0: 1.7-inch block micro-computers that interact with each other. We programmed the cubes using the Sifteo SDK and C++. The current prototype supports 8 Sifteo cubes. The tabletop application is implemented on a SUR40 device using the Microsoft Surface 2.0 SDK and written in C#. We detect the position and orientation of the Sifteo cubes and passive tokens on the tabletop using Byte Tags. We implemented Client-Server communication between the SUR40 and Sifteo applications using PyUSB, a Python module that supports USB access. Results are generated using the domain-specific programming language, Eugene [3].

Preliminary Evaluation

We conducted a qualitative study with 18 users (16 female, age 18-32) to evaluate the strengths and limitations of constructing rules and exploring a solution space using tangible interaction. One of the users is a primary investigator in the area of synthetic biology, three are iGEM students who did not participate in the previous study, 8 are undergraduate students majoring in life sciences, and 8 are undergraduates majoring in computer science. 12 of the users had some prior experience with the Eugene programming language.

Studies were conducted in the HCI Lab with users working in pairs (total of 9 dyads), sitting side by side at the tabletop surface. The study began with a warm up task that included an introduction to logic gates and required users to construct two different logical statements by combining active and passive tokens. Users were given all the physical tokens available and were asked to choose the pieces that made the most sense to them in order to build the logical construct. Following the completion of this task, users were given a brief oral and written introduction to synthetic biology, SBOL notation, the Eugene language, and the mapping between our physical tokens and Eugene. The study moderator also demonstrated how to use the physical active tokens in order to specify the design of a biological construct described in an SBOL diagram. As the experimental task, users were given two SBOL diagrams and were asked to construct the bio-design rules of the described biological constructs. Each design included between 5 and 10 biological parts and required specifying between 13 and 17 rules. This task mirrors a real-world iGEM task similar to the one used in the evaluation of our multi-touch Eugenie prototype.

Sessions were videotaped and notes were taken. Users were asked for their initial impressions following each of the two tasks and answered additional questions at the end.

Results

On average, dyads spent 36.5 minutes on the task (SD=9.3, MAX=50, MIN=23). All users were able to easily browse through a library of parts using the *tilt* gesture, and to find and select the parts that they needed. In general, users were able to figure out which operator tokens to use based on their physical form factor. Users easily identified the function of logical operators (e.g. AND, OR, <, >, =), but some found it difficult to identify the ALL and THEN operators. Groups that took longer spent more time discussing pieces and did not find some of the representations as obvious as others.

All dyads were able to specify the correct design rules for the biological devices, but in 6 cases, users attempted to use the tokens in ways different from the designed interaction syntax. For example, 4 users put the NOT operator on the side of the negated block rather than on the top, attempting to create a linear representation of the rule. 1 pair tried to rotate for REVERSE rather than turning the piece upside down, and 1 pair tried to use the = sign to specify THEN. However, in most cases (except the reverse), users were guided by the physical constraints to discover their misuses before they finalized the rule.

All pairs collaborated effectively with both members contributing physical and verbally to the task, using words, gestures, and pointing to express ideas. Users almost always designated a common space in which to place the rule in progress. 6 of the pairs lined pieces up in this common area, discussing the proposed formation before making the appropriate connections.

We also found evidence that users used the physical tokens to think. 4 pairs deliberately sorted pieces based on similar shape and moved the pieces they were using closer to them. Users also held or pointed to the piece they were considering. For example, when one user was talking about a gene that needed to be repressed, they made a counter-clockwise gesture with their hand, a gesture that mimics our representation of the REPRESS relationship.

The study drew our attention to several limitations: (1) *Limited interaction space* – While the physical tokens could be manipulated above the surface in-air, almost all users restricted their interactions to the bezel, which limited the extent to which users were able to explore the pieces. We suspect that the flat and puzzle-like form factor afforded manipulation on a flat horizontal surface rather than picking up the blocks; (2) *Rule Persistence* – While some users appreciated how the flexibility of active tokens "…allowed me to have more options with less blocks", in general, users had a hard time understanding the metaphor of "stamping" a rule on the surface. After stamping a rule, it is transferred to the surface and displayed as a Eugene statement in the rules view. The pieces can then be deconstructed and active tokens can be reused through dynamic binding. We observed that many users wanted each Sifteo cube to represent the same part for the entirety of the session. Users also left the constructed rules on the surface for later reference. Users were more likely to point to the physical structure rather than the code representation on the surface. We think that providing users with a *visual* rule representation on the surface, rather than just code could bridge the gap between the physical and digital representations, thereby helping users view stamped rules as persistent; (3) *User expectations* – While the use of physical constraints helped users to figure out syntax, user's expectations sometimes restricted their exploration of the pieces. Many users expected a more direct mapping between the physical syntax and natural (or programming) language. For example, we provide a "then" token that takes two arguments and represents "a→b". While users could easily identify the token, they also looked for an "if" piece even though it is not necessary syntactically. Several users also attempted to arrange the tokens linearly and read the rule from left to right like a sentence.

DISCUSSION

In this paper, we describe the design, development, and evaluation of two different implementations of Eugenie, a system for supporting novices in the bio-design process. While both implementations have limitations, they demonstrate the challenges and opportunities of applying TEI approaches to the exploration of large data sets through the construction of complex queries.

Some of the *challenges* highlighted by these case studies include: (1) Providing users with *rich representations* that integrate multiple facets of information (e.g. properties, structure, behavior) and are easy to manipulate. We found that in multi-touch interaction, visual clutter, target size, and occlusion could limit the effectiveness of visual representations; (2) *Closing the gap between tangible and digital representations*. We found that transforming tangible representations to digital requires visual representations that can be easily mapped back to its tangible counterpart, in order to reinforce persistent and continuous representation; (3) Supporting *aggregation* of complex configurations of digital and tangible representations into simpler representations. We learned that users want to combine complex structures representing a set of rules into a single reusable representation.

These case studies also highlight *opportunities* for TEI manipulation of large datasets: (1) *Achieving scalability* through the use of *active tokens* and *dynamic binding*. One of the core limitations of tangible user interfaces is scalability [16]. Providing tangible representations for large data sets can require a large number of tangibles that take up space and are difficult to manage. However, our tangible implementation of Eugene demonstrates that, by using active tangible tokens that can be dynamically associated by users, we can enable users to search and select subsets from a large data set (in our case, a data set of more than 3000 biological parts) with a compact set of tangibles; (2) *Utilizing new interaction spaces*. The tangible implementation of Eugenie supports interaction *on*, *in-front-of*, and *next-to* an interactive surface, by utilizing the surface bezel. In addition, active tokens can be manipulated independently of physical constraints through gestures. Designing for interactions beyond the surface helped to overcome challenges common to data-intense applications (e.g. finger size, occlusion, and visual clutter). In our study, we found that the interaction space on the bezel was too small (4 in.) to encourage exploration; this suggests a need to further consider how best to utilize the space *in-front-of* and *next-to* the surface; (3) *Enforcing interaction syntax with physical syntax*. Constructing complex queries often requires abiding to strict syntax. Physical constraints afford certain actions while preventing (or increasing the threshold for) others. We observed that, when using the tangible implementation of Eugenie, physical constraints allowed users to determine syntactic rules as well as correct errors; (4) *Thinking through action*. The Eugenie tangible user interface allows users to apply various strategies for reducing cognitive workload. For example, we observed that users sorted and arranged the tangibles around the surface, gestured with and pointed to the tangibles, and attempted various spatial configurations; and (5) Supporting *effective collaboration*. Both case studies support co-located collaboration. The multi-touch implementation provides multiple access points through parallel workspaces, allowing users to reinforce their territory or share a particular workspace. The tangible implementation provides multiple access points through the physical tokens. Both interfaces enable users to divide the problem spatially and temporally so that users can assume different roles or work together.

CONCLUSION AND FUTURE WORK

In this paper, we presented four main contributions: (1) lessons from the iterative design, implementation, and evaluation of a multi-touch and tangible interface for querying large data sets; (2) new tangible and scalable interaction techniques for using active tokens to navigate large hierarchical data sets; (3) a tangible interaction language for forming complex queries; and (4) a discussion of challenges and opportunities for applying TEI approaches to the exploration of large data sets.

Future work will include design iterations of Eugenie++ and comparative studies with isomorphs of Eugenie. We also intend to deploy and study Eugenie++ in an introductory synthetic biology course.

ACKNOWLEDGMENTS

We thank Traci Haddock, Swapnil Bhatia, Ernst Oberortner, Douglas Densmore, and the BU iGEM team. We also thank the undergraduate students that contributed to this work: Consuelo Valdes, Tiffany Chen, Catherine Guo, and Sravanti Tekumalla.

REFERENCES

1. (n.d.). Retrieved from Sifteo: https://www.sifteo.com/
2. Antle, A. N., & Wise, A. F. (2013). Getting down to details: Using theories of cognition and learning to inform tangible user interface design. Interacting with Computers, 25(1).
3. Bilitchenko L, Liu A, Cheung S, Weeding E, Xia B, Leguia M, Densmore D (2011). Eugene—a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. PLoS One 6(4).
4. Block, F., Horn, M. S., Phillips, B. C., et. al. (2012). The DeepTree Exhibit: Visualizing the tree of life to facilitate informal learning.Visualization and Computer Graphics, IEEE, 18(12).
5. Camarata, K., Do, E. Y. L., Johnson, B. R., & Gross, M. D. (2002). Navigational blocks: navigating information space with tangible media. In Proc. of IUI, ACM.
6. Drucker, S., Fisher, D., Sadana, R., Herron, J., & Schraefel, M. C. (2013). TouchVix: A Case Study Comparing Two Interfaces for Data Analytics on Tablets. In Proc. of CHI, ACM.
7. Glenberg AM (2008) Embodiment for education. In: Calvo P, Gomila T (ed) Handbook of cognitive science: an embodied approach, pp 355–372
8. Hornecker, E., Marshall, P., Dalton, N. S., & Rogers, Y. (2008). Collaboration and interference: awareness with mice or touch input. In Proc. of CSCW, ACM.
9. Isenberg, P., Isenberg, T., Hesselmann, T., Lee, B., Von Zadow, U., & Tang, A. (2013). Data visualization on interactive surfaces: A research agenda. Computer Graphics and Applications, IEEE, 33(2), 16-24.
10. Jetter, H. C., Gerken, J., Zöllner, M., Reiterer, H., & Milic-Frayling, N. (2011). Materializing the query with facet-streams: a hybrid surface for collaborative search on tabletops. In Proc. Of CHI, ACM.
11. Klum, S., Isenberg, P., Langner, R., Fekete, J. D., & Dachselt, R. (2012). Stackables: combining tangibles for faceted browsing. In Proc. of AVI. ACM.
12. Morris, M. R., Brush, A. B., & Meyers, B. R. (2007). Reading revisited: Evaluating the usability of digital display surfaces for active reading tasks. In Proc. of ITS, IEEE.
13. Morris, M. R., Fisher, D., & Wigdor, D. (2010). Search on surfaces: Exploring the potential of interactive tabletops for collaborative search tasks. Information processing & management, 46(6), 703-717.
14. Patten, J., & Ishii, H. (2000). A comparison of spatial organization strategies in graphical and tangible user interfaces. In Proc. of DARE, ACM.
15. Rekimoto, J., Ullmer, B., & Oba, H. (2001). DataTiles: a modular platform for mixed physical and graphical interactions. In Proc. of CHI, ACM.
16. Shaer, O., & Hornecker, E. (2010). Tangible user interfaces: past, present, and future directions. Foundations and Trends in HCI, 3(1–2).
17. Shaer, O., Mazalek, A., Ullmer, B., & Konkel, M. (2013). From Big Data to Insights: Opportunities and Challenges for TEI in Genomics. In Proc. of TEI. ACM.
18. Shaer, O., Strait, M., Valdes, C., Wang, H., Feng, T., Lintz, M., ... & Liu, S. (2012). The design, development, and deployment of a tabletop interface for collaborative exploration of genomic data. International Journal of Human-Computer Studies, 70(10), 746-764.
19. Synthetic Biology Open Language (2011) Retrieved from http:// www.sbolstandard.org/
20. Thudt, A., Hinrichs, U., & Carpendale, S. (2012). The bohemian bookshelf: supporting serendipitous book discoveries through information visualization. In Proc. of CHI, ACM.
21. Ullmer, B., Ishii, H., & Jacob, R. J. (2003). Tangible query interfaces: Physically constrained tokens for manipulating database queries. In Proc. of Interact.
22. Valdes, C., Ferreirae, M … & Shaer, O. (2012). A Collaborative Environment for Engaging Novices in Scientific Inquiry , In Proc. ITS 2012, ACM.
23. Valdes, C., Eastman, D., Grote, … & Konkel. M. (2014). Exploring the Design Space of Gestural Interaction with Active Tokens through User-Defined Gestures. In Proc. of CHI 2014, ACM.
24. Voida, S., Tobiasz, M., Stromer, J., Isenberg, P., & Carpendale, S. (2009). Getting practical with interactive tabletop displays: designing for dense data, fat fingers, diverse interactions, and face-to-face collaboration. In Proc. of the ITS, ACM.
25. Xu, W., Chang, K., Francisco, N., Valdes, C., Kincaid, R., & Shaer, O. (2013). From wet lab bench to tangible virtual experiment: SynFlo. In Proc. of TEI. ACM.
26. Zigelbaum, J., Horn, M. S., Shaer, O., & Jacob, R. J. (2007). The tangible video editor: collaborative video editing with active tokens. In Proc. of TEI, ACM.