Stackables: Combining Tangibles for Faceted Browsing

Stefanie Klum[1] Petra Isenberg[2] Ricardo Langner[1] Jean-Daniel Fekete[2] Raimund Dachselt[1]
[1]User Interface & Software Engineering Group
Otto-von-Guericke University, Magdeburg
stefanie.klum@steelis.net (rlangner | dachselt)@isg.cs.uni-magdeburg.de
[2]Team Aviz
INRIA Saclay
(petra.isenberg | jean-daniel.fekete)@inria.fr

ABSTRACT

We introduce *Stackables*: tangibles designed to support faceted information seeking in a variety of contexts. We are faced, more than ever, with tasks that require us to find, access, and act on information by ourselves or together with others. Current interfaces for browsing and search in large data spaces, however, largely focus on the support of either individual *or* collaborative activities. Stackables were designed to bridge this gap and be useful in meetings, for sharing results from individual search activities, and for realistic datasets including multiple facets with large value ranges. Each Stackable tangible represents search parameters that can be shared amongst collaborators, modified during an information seeking process, and stored and transferred. We describe Stackables, their flexible and expressive combination to formulate queries, and the underlying interaction concept in detail. An evaluation provides initial evidence of their usability in targeted and exploratory information seeking tasks.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input Devices and Strategies*

Keywords

Tangible UIs; faceted browsing; faceted search; collaboration

1. INTRODUCTION

Imagine a medium-sized business has to decide on the purchase of computer equipment for its employees. The decision on what to purchase can have profound impact on employee productivity and business profitability and choices on options and specifications are therefore not easy to make. Typically, decision makers from several departments with varying priorities and skills are involved in the selection process. Faceted browsing can help decision makers: they can consider data from different conceptual dimensions and incrementally refine a data view by restricting facet values such as price, technical attributes, or ratings (e. g., [6, 12]).

Figure 1: Stackables are small tangibles with display and two control wheels to adjust facet values. They can be combined to filter a complex faceted data set.

Yet, even given a complete dataset of equipment options, the scenario above poses a number of challenges to regular faceted information seeking interfaces: (a) The decision makers will likely begin their investigation with a private exploration and selection. Afterwards, the results of this exploration, as well as facet values which led to them, have to be shown to the others and discussed. Mechanisms are, thus, required to share not only result sets but also facet values. (b) During a decision meeting, the shared use of an information seeking interface can help to advance the discussion and clarify options. Such an interface needs to be easily set up and allow participants to fluidly share, transfer, and manipulate facet values while working in a closely or loosely coupled fashion [10]. Thus, an information seeking interface to support this scenario needs to be easily used in individual as well as collaborative information seeking and allow for an effortless transition between both.

The above scenario and shortcomings of current search systems have been the primary motivation for our work. We address the challenge of shared faceted browsing where small groups have to make a decision on items described by facet categories. This challenge is very generic; it happens in organizations such as companies, schools, or even at home when selecting vacation sites or items to buy. It often involves an initial step where individual preferences are made, then shared with a group to reach a decision,

AVI '12, May 21–25, 2012, Capri Island, Italy

followed by discussion and further individual or collaborative refinements.

In this paper, we introduce *Stackables*: physical widgets designed to be usable for faceted browsing by individuals, as well as groups of people in a co-located setting. Stackables are tangibles that can store search queries, can be manipulated, shared, transferred, and used in the negotiation of results and search goals with multiple other people.

We discuss related work in Sec. 2, introduce Stackables and their interaction concept in Sec. 3, and the final realization in Sec. 4. The user study in Sec. 5 provides evidence of the usability of our concept and design. We conclude with a Discussion of the use of Stackables for the proposed scenario in Sec. 6 and give pointers to future work.

2. RELATED WORK

How to support information seeking activities in large amounts of information is an active research area. We first review GUIs related to our approach and then discuss related research on tangible user interface (TUI), with a focus on TUIs for information seeking.

GUIs for Faceted Information Seeking. Faceted Browsing is a fundamental information seeking paradigm that has become common in online shops and similar applications. GUIs have been proposed as early as 1993. In Filter/Flow [19] Young & Shneiderman describe facets as water filters that narrow down a stream of results.

The core of the technique has remained, but has been extended in research. Many existing GUIs use a stacking metaphor and thus relate to our *physical* stacking for query assembly. FacetLens [12], for example, uses faceted browsing to filter results, shows available facet values for drill-down, and gives an information scent on the number of remaining data values after another refinement. The filter history in FacetLens is shown as a stack of the spatial hierarchy associated with it. Stacking is similarly shown in FacetZoom [6]—a faceted browsing interface for hierarchical facets. Single facets are represented as space filling trees and displayed in a widget at the bottom of the screen. FacetMap [15] also shows multiple facet hierarchies and related data items according to a space-filling layout, but in a scalable, joint visualization space. In contrast, we wanted to decouple facet widgets from the result space for more flexibility in collaboration. FindFlow [9] supports the construction of faceted searches using a graph visualization of the data flow. Multiple searches can be conducted in parallel. These form different paths along the search graph so information seekers can compare them interactively and even join several subqueries. FindFlow supports interactive query refinement, manipulation, and adaptation which influenced our interaction design.

TUIs for Faceted Information Seeking. Early in the development of TUIs, graspable bricks [8] were proposed to physically control parameters and as tokens for representing virtual data. Navigational Blocks [5] are an early tangible faceted browsing interface in which physical blocks form simple database queries. Each block represents one query facet, its six faces the possible values. In contrast, our goal was to support a large number of facet values. Ullmer et al. [18] generalized the idea and introduced two tangible interfaces to form database queries—one employing parameter wheels for fixed search facets, the other utilizing parameter bars that can be dynamically assigned to facets. As in our system, the tangible query interface supports both discrete (e. g. building types) and continuous (e. g. price) parameter values. Since the tokens are physically constrained and require query racks, they are less portable and less suited for collaboration. In follow-up work, the Cartouche concept [17] was used as a generalization of tangible menus. While very flexible and combinable with a number of technical setups and applications, the overall approach is likely too technical for everyday collaborative applications. FacetStreams [11] is the most closely related system to ours. Here Filter/Flow [19] principles are applied to a multiuser scenario [11]. The application is a hybrid interface which uses passively tracked tokens with tabletop feedback and multitouch interaction. A glass token represents a search facet that can be combined with others in a query "stream." Glass tokens can be quickly added, removed or repositioned on the table surface during exploratory search. Developing a similar fluid interaction was also very important to us and one main reason for basing our solution also on tangibles.

After the review of some important GUI and TUI solutions it can be, thus, concluded that faceted browsing interfaces that bridge the gap between individual and collaborative information seeking are still very rare.

Tangibles that Form Stacks. Most tangible-based query interfaces use a *horizontal* arrangement [5, 11, 13, 19]. Yet, tangible systems and hardware solutions have also been proposed to support vertical stacking in contexts other than information seeking. Lumino [4] uses stack-detection technology based on an interactive tabletop with a camera beneath the surface. The Luminos tangible blocks possess individual markers and are filled with fiber bundles. The stacking height is limited by the resolution of both the camera and the fiber bundles (no more than five layers). In Stacks on the Surface [3], tangibles are placed upon each other to form a composite image. These composite images can be recognized by common tabletop technology. However, stack height is constrained to a few millimeters above the table (5mm in experiments). Consequently the tangibles used in this approach are made from slim sheets of plastic. Both hardware solutions are not suited for our scenario, as our goal was to develop a system that could be used independently of the availability of a tabletop display.

3. STACKABLES: PILED TANGIBLES

Several benefits of tangible interfaces made them a promising option for our application scenarios. Tangibles allow parallel interactions in a co-located setting such as a meeting, tangible facet representations visualize queries in a physical form (cf. [18]), and allow all participants direct input. Finally, tangible query representations are a form of physical manifestation of the result of an information seeking *process* that can be stored and re-accessed for further work in other settings. Inspired by related work, our goal was to constrain the tangibles' footprint and support functional composition by stacking. In contrast, we did not only want to have a representation of values, but the ability to change them with the same tangible. Thus, similar to many GUIs for faceted browsing, Stackables' query construction is based on a *vertical* stack. Our design of Stackables differs from FacetStream [11] by supporting an interplay of work in individual and collaborative spaces, vertical stacking, and a separation of input and output space—we do not require the use of a tabletop display. This also differentiates our work from many other predecessors, such as DataTiles [13].

Throughout the remainder of the paper, we use a scenario—based on a dataset of 1580 books—to illustrate our requirements and to discuss our interaction concept with examples.

3.1 Motivating Scenario and Requirements

In the course of a school event, representatives of teachers, students, and parents choose text books from a database of available literature. In a *preparation* phase each representative collects preferences from his/her group and conducts preliminary research alone. The results are taken to the meeting (e. g., on a laptop or as a printout). During the *meeting* phase all representatives first need to show each other's results to (likely) realize that preferences are scattered and a compromise has to be found. They now engage in phases of collaborative work to find alternatives. Search paths and facet values have to be transferred to a shared PC manually. No parallel work is possible as only one mouse is available. In the *result* phase, the representatives reach a result after several rounds of discussion and need to take it back to the groups they represent. Like in the first phase, results will be somehow printed or copied.

The above scenario includes several challenges for interface design in terms of work styles, saving of queries and results, and decision making. These challenges lead directly to our design requirements. Our goals were to support:

R1 realistic datasets (large number of facets/facet values),
R2 faceted browsing principles, simple boolean queries,
R3 separation of search input and result output—to allow for Stackables to be used in regular meeting spaces,
R4 use in both individual and collaborative work phases.

3.2 Design and Interaction Concept

Stackables consist of two main components: facet tokens and ground plates. Facet tokens as the main building blocks are stacked to form queries. Stackables also require a computer—to analyze and issue the search queries built with the facet tokens—and an output display that shows the results of one ore multiple parallel queries. With this separation of input and output space, we satisfy R3 and allow R4.

Facet Tokens A Stackable token can represent any numeric or categorical facet in our dataset and can be assigned and re-assigned on the fly. Thus, each Stackable is generic and contains information about all facets and their values.

Ground Plates Ground plates hold no facets, but are the starting point of every stack. They provide stability and help to distinguish tangibles that form stacks and those that are merely placed on the surface, outside of a query. Ground plates are a design concept of secondary importance to query formulation and our evaluation. We, thus, do not focus further on their design and implementation here.

3.2.1 Appearance

We designed the appearance of Stackable tokens according to the following visual and haptic properties:

Shape All tokens share a similar abstract box shape that can be easily stacked. To support vertical stacking we chose the box shape to be more wide than high. Facet type is not encoded permanently on the box to allow for reuse. Instead, a *colour-display* on the front shows facet data. On the front and back *two turnable wheels* allow for selection of facets and facet values. The display and wheels are positioned off-center so that the up-direction of a token is readily visible. This also gives room for a button-like area on the left.

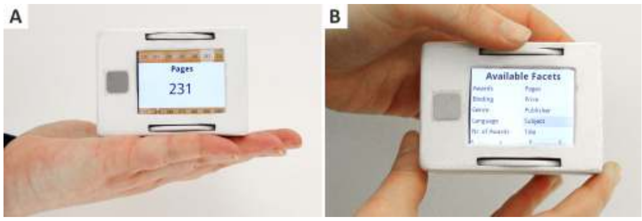

Figure 2: Stackables size related to a female hand (left) and when held between two hands (right, available facets in the book dataset are shown).

Color The token's box is held in a neutral white to further emphasize its versatility in facet choice (R1). On the display, each facet name is printed and highlighted by a distinctive color hue to make the facet choice visible from a distance.

Material The material of the box was chosen to allow for ephemeral personalization. Query states can be stored within each Stackable and washable colours or stickers can be used to name and tag a tangible on the outside. This is useful as the token is usually turned off when not in use.

Size The display requires a minimum size and resolution to be readable across a meeting room table. For portability, tangibles should be small and easy to carry. In our final design, we chose a trade-off in transportation comfort and interface usability. Facet tokens are sized so that they can be held during interaction but also remain stable when manipulated within a stack (see Fig. 2).

Display *Categorical facet values* are alphabetically sorted and represented as a two-column list (see Fig. 3, A/B). The top of the list shows the facet name. Below the list, a quick-jump context widget shows the starting letters of available facet values. For *numerical facets*, the middle section shows the currently selected value(s) and facet name. On the top, a range of subsequent available numbers is shown. The bottom row again shows a quick-jump context widget of the whole range of values divided into bins (see Fig. 3, C/D). The top turnable wheel selects single values, the bottom wheel navigates the quick-jump menu. After ten seconds of inactivity the display changes to a simple facet name and value representation (see Fig. 4, right).

3.2.2 Interaction

Five operations can currently be performed on Stackables, but extensions are easily possible. For the five interactions, the two wheels, the button, and token orientation are used.

Facet Selection and Deletion When no facet has yet been chosen for a token, the display shows the available facets of the loaded dataset. These facets can be navigated with the wheels and a selection is made using the button (Fig. 2, right). Once a selection is made, the display shows the available facet values. To reset a token, the button must be held for three seconds. An alternative design involves shaking the token to reset it—however, this interaction is only possible when the token is not part of a stack. With this interaction design we meet R1 in that an arbitrary number

Figure 3: Displays of Stackable tokens for two categorical (A, B) and two numerical (C, D) facets.

of facets can be picked from a dataset.

Single-Value Selection We designed the interaction to allow for the selection of facet values in arbitrarily large facet ranges to further meet R1. The top wheel performs single steps through facet values while the bottom wheel uses larger steps (see the "Display" paragraph), akin to the Alphaslider [1]. Single facet values can also be saved by pressing the button to allow for logical OR connection of facet values. We favor this encoding over a physical representation (e.g. by horizontal grouping) as it fullfills (R2) without complicating stack construction.

Range Selection Range selection is activated by twisting the top and bottom wheel apart at the same time. The screen of the token always shows the currently selected beginning and end values of a range, as well as a number of values in-between. During range selection the bottom wheel steps through facet values one-by-one so that the beginning and end values of a desired range can be exactly specified. Range selection is de-activated once the top and bottom wheel again point to the same facet value. Range selections, like single-value selections, can be saved by pressing the button. Through the save function both single-value and range-value selection can be combined (R2). To simplify the interaction, particular values or ranges of a combined query cannot be deleted separately. Instead, all selected facet values are deleted at once (see above).

Query Construction Multiple facet tokens can be combined with a logical AND through vertical stacking (Fig. 4). The display shows the availability of facet values given a selection further below in the stack. For example, if a "subject" has been stacked on top of a "genre" facet, the subject facet will highlight only those subjects in which books of the selected genre have been published (see Fig. 4). Facet value selection can be changed for tokens placed within and outside of a stack. In our example, choosing a different genre for the facet at the bottom of the stack will not remove the selected subject in the "subject" token above—even if it is no longer available. This is done to avoid accidental changes of a large selection stack. Each successive turn of one of the wheels in higher tokens (here the "subject" token) will, however, automatically jump only to the next available facet value. This allows speeding up query construction and minimizes the risk of empty result sets. When a facet token is taken out of a stack for selection, it is treated as a non-active token until it is either replaced in the stack, placed on another stack, or placed elsewhere on a surface outside of a query. All facet values for this token are set to "available" but all previous selected values and saved selections still remain. This addresses R4 as collaborators can work in parallel on their own tokens and stacks.

Figure 4: Two facet tokens in a stack. Left: The top showing the available values in black and the unavailable values in grey. Right: The same facet values after a period of inactivity.

.

Figure 5: Result representation of a query with three Stackables in a single stack (A). B.1 shows the cover of alls books matching all selected facets values in the stack, B.2 shows partial matches, and B.3 all books matching no selected facet value.

Negation Negation of a selection can be achieved through a combination of "saves" of single and range-facet values or by simply placing a Stackable token upside-down in a stack. The display rotates to show the values in correct orientation. All available facet values other than the currently selected ones now take part in the query.

Together, single-value selection, range selection, stacking, and negation support R2 through the basic principles of faceted information seeking and simple boolean queries. Specifically, we support the following faceted search interactions [14]: zoom-in (making the query more specific), zoom-out (generalizing the query), shift (replacing part of the query), slice and dice, and range selection. In earlier prototypes we included interaction concepts for Pivot and query-by-example through a "query copy" into an empty

Figure 6: Five design iterations of Stackables. From oldest on the left to the final design on the right.

token and a follow-up touch-screen interaction. We finally decided to first evaluate the more basic faceted browsing interactions with our final prototype.

Result Representation Finally, the output of one or multiple queries is shown on a display or projection—whose size is typically chosen depending on the number of participants in the search process. The result representation consists of two main parts: a representation of the current stacks formed by the participating Stackables (Fig. 5, A) and a representation of the results of the queries (Fig. 5, B). The stack visualization mimicks the physical stacks—facets are displayed in the same way and color as on the token displays. The result representation shows three categories of results: perfect matches which satisfy all participating queries (all formed stacks), partial matches (items matched by only some stacks or parts of a stack), and all unmatched items. Fig. 5 shows the results of a one-stack query. It can be extended to also show multiple stacks and which facets and stacks match which book. These extensions are important for collaborative search tasks. As the visualization of collaborative search results is a research task on its own, the collaborative visualization, its requirements, and design will not be described in detail here. While search participants interact with the tangibles and reorder the stacks, the visualization updates immediately to show new results and near matches. This behaviour allows people to actively explore a dataset and adjust queries on the fly.

4. IMPLEMENTATION

During the development of Stackables, we built a series of five prototypes (Fig. 6) during which we experimented with different hardware, materials, and stacking solutions. Fig. 7 summarizes the final Stackables system design with communication channels and participating components. A desktop application (implemented in C#/WPF) loads the data, handles incoming queries, and renders a dynamic result representation. The Stackable tokens consist of two separate parts: a mobile phone with software to display facets and their values (written in Java & Android SDK) and a small Arduino FIO micro-controller [2] of the open source Arduino project. The controller reads sensor values from the two turning wheels and sends them to process to the desktop application. The desktop application performs the dynamic query and sends the results back to the phone.

4.1 Hardware Setup

Components: The mobile phone and micro-controller are hosted in a wooden case, designed to comply to the appearance guidelines of Sec. 3.2.1. Fig. 8 shows both parts in an open case. The FIO micro-controller fits our needs well due to its small form factor and included connectors for a rechargeable battery and an Xbee wireless modem [7]. The Arduino is connected to two rotary encoders. Both were modified to move smoothly without noticeable stepping.

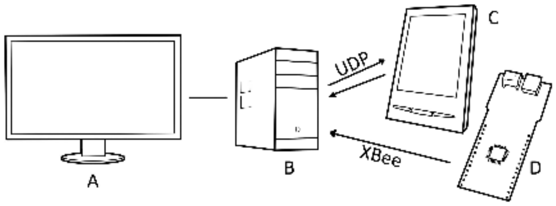

Figure 7: Schema of the interaction between system components: A central computer (B) renders a result visualization (A). An Arduino (D) sends sensor input via XBee wireless. The Android phone (C) communicates via UDP.

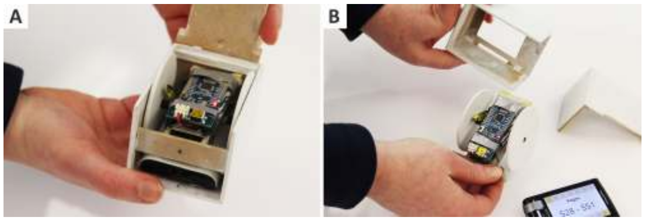

Figure 8: Assembly of the prototype components.

Sensor information processed by the Arduino is transmitted to the main application through an XBEE wireless network. We used a Sony Ericsson Xperia X10 mini smart phone [16] rather than the micro-controller to provide real-time visual feedback of query selection. The phone is equipped with a 2.6" color display (320 × 240 pixels resolution), a good tradeoff between small size and readability. The case is flexible enough to indent slightly, so pressing down on the active area next to the display triggers a button event on the mobile phone. Communication between the cell phone and the desktop application uses UDP with an ad-hoc protocol.

Stacking Solution: We had to develop new stacking methods, as most recent stacking solutions (Sec. 2) rely on the availability of a tabletop display. We experimented with two alternatives. During early iterations, we detected stacking through force-sensitive resistors that register the changing weight of tangibles stacked above it. Every token was able to detect three layers of stacking. However, we saw a major drawback in our tests: a lack of reliability when users rested their hands on a stack. For our latest design we favor magnetic reed switches that do not require direct contact to detect a token neighbor. Four of these sensors are positioned in a row at the top and bottom of each tangible. On the opposite side, four slots hold small magnets, triggering the switches when tangibles are stacked. A drawback to this approach is its reliance on proper alignment. To aid users in forming stacks, tangibles require physical constraints ensuring correct positioning. Our experiences with prior iterations of Stackables shows that users prefer to tilt and angle stacked tangibles in a variety of ways. Another technical solution for a next design could be the use of near field communication (NFC).

4.2 Software Implementation

Every Stackable is operated by two pieces of software: One component runs on the Arduino to collect and send sensor values, the other operates the result representation. Both are coordinated through the central application that also handles query interpretation. Once the application interpreted the incoming sensor values, selected facet and facet values are passed on to the Android phone. A corresponding facet value representation is computed directly on the mobile phone, depending on facet type and the number of selected items. Facet values are highlighted when selected. During range selection the list is split into two columns when start and end points of the selected range no longer fit on one screen. Fig. 3 shows four screenshots from the final implementation.

5. EXPERIMENT

In order to evaluate our design concept and its implementation we had two options: either study the design and query formulation in detail or opt for a more encompassing collaborative study similar to our scenario. We chose the first option and evaluated query formulation and interaction with single participants with our latest prototype—before building more Stackables for a larger study. One goal of the study was to find ways to refine and improve the design as well as validate our fundamental concepts. We chose not to conduct a comparative evaluation but consider doing so in the future in the context of a larger collaborative user study.

5.1 Participants

Twelve unpaid participants (8 male/4 female) participated in the experiment (average age: 27). The participants had mixed experience with faceted browsing. Five reported familiarity with the term and two experience with faceted search. However, all reported past usage of online shopping portals which allow query refinement through facets (e. g., ratings, price range). All reported to at least sometimes refine their queries through categories in these shopping portals and nine reported to do so in 50% of their searches or more. All participants were daily computer users.

5.2 Design and Procedure

During the experiment, participants sat at a desk and manipulated up to three Stackable tokens from our latest design iteration. A 24" LCD (1920×1200 pixel resolution) was placed 60 cm in front of them and displayed the result representation. The study software was driven by a laptop equipped with an Intel i7 quadcore (2 Ghz), 8 GB RAM, running Windows 7. Each participant completed five tasks with increasing difficulty. Before each task, participants received verbal instructions from an experimenter, were allowed to interact with the Stackables, and also received several training trials. On average the study lasted 75 minutes. Participants were encouraged to share thoughts and ideas on the interface at any time during the experiment.

5.3 Tasks

The study consisted of the following tasks in order:
Task 1: tested how well participants understood the usability of selection wheels in conjunction with the token display. Using one Stackable token, participants completed eight single selection trials, four with the categorical and four with the numerical interface. The result representation was disabled in this trial. To reduce user fatigue, a facet value was pre-selected on each token at the start of each trial in this task and the following three.
Task 2: tested the interplay of single-selection and result representation. The selection tasks required participants to look at the result representation. An example question was: “Select a year between 1850 and 1900 with 2 books, and none of these books show words on the cover.” Participants completed eight trials, four with each interface.
Task 3: tested range selection. Ranges were 10 facet values long and were positioned around the 20th or 160th facet value. Participants completed 8 trials, 4 with each interface.
Task 4: introduced stacking. Participants completed four trials in which they selected either a range or single values on one of two tokens and combined them in a stack. For this task we collected mainly qualitative data.
Task 5: was an exploratory browsing task of one trial in which participants could combine the previously experienced features freely. We collected mainly qualitative data.

5.4 Data Collection and Analysis

During each task we created time-stamped log entries for every interaction with the Stackables and recorded video- and audio-data. An examiner was present for each experiment and took notes of participants’ comments and conducted a semi-structured interview after each task. A post-task questionnaire gathered feedback on the interactions introduced in each task. The questionnaire and log files were analyzed using descriptive statistics (distributions & central tendency) while the notes were aggregated to build higher-level categories of comments from participants. A log visualization was built to extract information on participants’ strategies to design queries. Video data was used to validate results and inform us in case a participant exhibited unusual behavior in log files. The results of the questionnaire are reported below together with our observations, task completion times, and errors made.

5.5 Results

In the following we step through the result in the order of tasks presented to participants.

Task 1 – Single Value Selection: Participants completed trials on average approximately equally quickly with the categorical (20 s) and numerical (16 s) interfaces. Participants only made errors on $\approx 4\%$ of trials. In the questionnaire 11/12 participants reported to have found single selection easy to use (one was undecided) and all found it easy to remember. Seven participants, however, suggested to improve the context information on the tangible display. Five asked for a preview of values a turn of the bottom wheel would select next, two mistook the intermediate context values as such. Nine participants found the bottom wheel to be helpful, only one participant reported not to like it.

Task 2 – Single Values, Eyes-Free Interaction: Average completion time for one trial was 45s. We observed that participants spent little time reaching the area of interest in the data compared to how long they spent validating the task conditions. The task required participants to observe the changing results closely. During the experiments we observed two different strategies: four participants glanced mainly at the Stackable until they reached the required value range, then switched their attention to the result representation, but continued to glance down at the token in their hands (up to five times per selection). The other eight participants also navigated to a starting point, and then concentrated on the visualization and only glanced down occasionally or before committing their result (zero to two times per selection). Nine participants reported to like the result representation on a separate screen. During the experiment, two participants particularly praised the ability to work eyes-free. Two others asked for additional haptic or audible feedback to know when the next value was selected without looking down at the Stackable at all. These observations show that Stackables are well suited for eyes-free interaction when fine-tuning a selection. When participants needed to make more elaborate adjustments, like finding a starting point in the overall value range, they preferred visual feedback directly on the tangible.

Task 3 – Range Selection: All participants found the concept of range selection easy to understand, the mode switch was clearly recognizable, and the selection was fairly fast. On average participants needed 56s to complete a specific range selection. Compared to the first task, completion time for selecting two boundary values should have averaged around 32s-40s. Task 3, however, showed several issues with our range interaction concepts. Twisting the bottom wheel before the top to enable the range selection mode made values jump and led to surprising selections. This problem will be solved with value caching in software. In addition, participants found it tedious that they could no longer skip values during selection. A spinnable wheel could solve this issue in the future. During the experiment three participants gave up on activating ranges using the gesture and were aided by the examiner. Two more participants required aid for some of the subtasks. After the examiner remotely activated a range selection, participants were able to adjust the boundaries without help.

Task 4 – Query Building: All participants were able to combine facets to form stacks and solve the trials. Ten understood the concept immediately, only two needed additional explanations. After completing the task, 11/12 participants found query building with Stackables logical and comprehensible (one was undecided). During the course of the experiment, three participants commented that they enjoyed building their query in a physical representation. Another four commented that they found the stacking metaphor easy to understand and liked to have a distinct tangible for every facet. After task completion only one participant found the virtual stack representation on the monitor helpful in building stacks. Perhaps it could be removed in the future. We observed two major strategies in query building. Four participants picked up each Stackable separately, selected the desired facet value, then placed it in a query. Five participants selected the first value holding a Stackable in their hands, but added the second token to their stack before adjusting its selection. Three participants alternated between both strategies, nobody opted to form a stack of two tangibles before selecting any facet values. Close attention was paid to the way participants positioned and manipulated tangibles inside a stack. The majority of participants neatly aligned their Stackables, but one person rotated the upper tangible by 90 degrees. When asked by the examiner, the participant reported that he did not want to accidentally touch a selection wheel on the lower token. We take this as an indication to rethink the position of the selection wheels in further iterations. Several participants criticized that Stackable displays were hard to read from their position. One participant circumvented the problem by angling upper stacked tangibles towards himself when adjusting facet values, one angled the complete stack in a similar fashion to be able to read both displays. One participant even picked up the whole stack, holding two tangibles at face level in his left hand while adjusting facet values with his right. In further iterations displays with a wider view angle or tiltable displays can relieve this problem.

Task 5 – Free Exploration: In this task, participants were asked to find 5–10 award winning books that could be of interest to children. Unlike Tasks 1–4, there was no right answer to the task. We gathered observations on three aspects of Stackables: participants' ability to apply the query metaphor freely, data access strategies, and participants' impression of Stackables' suitability for collaboration. The stacking metaphor was readily applied freely by all participants and 11/12 reported to find it easy to form the queries they were interested in. Seven participants asked for additional query building complexity: three wanted to form OR connections between facets (included in an earlier concept, but not in the study), four asked for an option to select a facet as a whole. These observations are encouraging for the development of a more elaborate stacking syntax for further tests. We expected participants to narrow down their selection until they roughly reached the 5–10 books asked for. Six participants (three of them female), however, selected a larger subset than needed and wanted to come to a decision based on book cover and other meta-data. Our current result representation did not show all the available meta-data but should be extended to support this search strategy. After completing the task, participants were asked if they could imagine sharing this interface in a group scenario. All twelve participants answered this question affirmatively. Several even made suggestions for possible group scenarios, confirming our intentional design for bridging the gap between individual and collaborative information seeking with Stackables.

6. DISCUSSION

Our results validated our Stackables concept and its suitability for faceted information seeking scenarios. We showed that Stackables are versatile and promising. They were tested for a dataset containing over 1500 books, nine facets with facet values of approximately 100–600 values in range. Participants readily operated the tokens both eyes-free and by looking on the token displays, seemlessly switching strategies according to the task at hand. Our stacking metaphor and the resulting query syntax were easily understood and applied by most participants. Though the experiments were not focused on testing affordances, our observations of stacking behavior confirmed that our hardware realization will support the intuitive stacking behaviour by the majority of users. Feedback on our result representation was positive. Despite the overall positive reactions to the Stackables concept and realization, our experiments also uncovered regions for improvement. For the result representation, participants asked for a result counter and additional detail information, and were less excited about the stack visualization currently provided. The majority of participants relied on the physical tokens as representation of their query, affirming the requirement of separated search input and result output (R3). Further experiments will show if a stack visualization is beneficial in more complex queries. Activating range selections proved to be difficult in our current prototype and needs to be re-designed in software.

We did not test all aspects of our scenario. In particular we opted not to include collaboration between users before validating our basic interaction concept. Still, results from both questionnaires and verbal feedback indicate that collaborative search is an important application to consider further. Our participants did not know about our design scenario but all envisioned the collaborative use of Stackables. Many saw the interface's main potential in collaborative settings and public spaces but could not imagine to use the tangible interface solely for individual searches. One particularly interesting suggestion for future applications is the use of Stackables as a rating interface. Combining ratings per token in one stack allows a rating task to benefit from the same aspects of flexibility and retained intermediate results we intended for our information seeking scenario.

7. CONCLUSION

Making information seeking both more efficient and simpler is an important research endeavour. We described Stackables as a new tangible solution for faceted information seeking and make three main contributions: we a) explore the feasibility of vertical stacking for query construction, b) provide a flexible interaction concept for large datasets with many facets and facet values, and c) constructed Stackables to be usable in face-to-face meetings and for individual use. A user study assessed the usability of our most recent prototype and provides insight on how participants created and combined queries by manipulating facet tokens and by combining them in a vertical stack. The results validated the interaction concept and showed that our requirements were well translated into design, but also showed areas for improvement. Taking the study results into account, we are working on constructing a larger set of Stackable tokens, with magnetic switches or NFC tags to track stacking, which can be used and tested in a full collaborative meeting scenario.

8. ACKNOWLEDGMENTS

This work was partly sponsored by the French Research Organization, project grant ANR-11-JS02-003.

9. REFERENCES

[1] C. Ahlberg and B. Shneiderman. The Alphaslider: a compact and rapid selector. In *Proc. of CHI*, pp. 365–371. ACM, New York, NY, USA, 1994.

[2] Arduino. Arduino.cc. Website, 2011. `http://arduino.cc/` (last accessed: Dec, 2011).

[3] T. Bartindale and C. Harrison. Stacks on the Surface: Resolving Physical Order Using Fiducial Markers with Structured Transparency. In *Proc. of ITS*, pp. 57–60. ACM, New York, NY, USA, 2009.

[4] P. Baudisch, T. Becker, and F. Rudeck. Lumino: Tangible Blocks for Tabletop Computers Based on Glass Fiber Bundles. In *Proc. of CHI*, pp. 1165–1174. ACM, New York, NY, USA, 2010.

[5] K. Camarata, E. Y.-L. Do, M. D. Gross, and B. R. Johnson. Navigational Blocks: Tangible Navigation of Digital Information. In *Proc. of CHI EA*, pp. 752–753. ACM, New York, NY,, 2002.

[6] R. Dachselt, M. Frisch, and M. Weiland. FacetZoom: A continuous multi-scale widget for navigating hierarchical metadata. In *Proc. of CHI*, pp. 1353–1356. ACM, New York, NY,, 2008.

[7] Digi. XBee Technology. Website, 2011. `http://www.digi.com/` (last accessed: Dec, 2011).

[8] W. Fitzmaurice, H. Ishii, and W. Buxton. Bricks: Laying the foundations for graspable user interfaces. In *Proc. of CHI*. ACM, New York, NY, USA, 1995.

[9] T. Hansaki, B. Shizuki, K. Misue, and J. Tanaka. FindFlow: Visual interface for information search based on intermediate results. In *Proc. of APVis*, vol. 60, pp. 147–152. ACS, Inc., Australia, 2006.

[10] P. Isenberg, D. Fisher, S. A. Paul, M. Ringel Morris, K. Inkpen, and M. Czerwinski. Collaborative Visual Analytics Around a Tabletop Display. *IEEE TVCG*, 18(5):pp. 689–702, 2012.

[11] H.-C. Jetter, J. Gerken, M. Zöllner, H. Reiterer, and N. Milic-Frayling. Materializing the Query with Facet-Streams: A Hybrid Surface for Collaborative Search on Tabletops. In *Proc. of CHI*, pp. 3013–3022. ACM, New York, NY, USA, 2011.

[12] B. Lee, G. Smith, G. G. Robertson, M. Czerwinski, and D. S. Tan. FacetLens: Exposing Trends and Relationships to Support Sensemaking within Faceted Datasets. In *Proc. of CHI*, pp. 1293–1302. ACM, New York, NY, USA, 2009.

[13] J. Rekimoto, B. Ullmer, and H. Oba. DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions. In *Proc. of CHI*, pp. 269–276. ACM, New York, NY, USA, 2001.

[14] G. M. Sacco and Y. Tzitzikas, eds. *Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience*. Springer, Germany, 2009.

[15] G. Smith, M. Czerwinski, B. Meyers, D. Robbins, G. Robertson, and D. S. Tan. FacetMap: A Scalable Search and Browse Visualization. *IEEE TVCG*, 12(5):pp. 797–804, 2006.

[16] Sony Ericsson. SonyEricsson.com. Website, 2011. `http://www.sonyericsson.com/` (last accessed: Dec, 2011).

[17] B. Ullmer, Z. Dever, R. Sankaran, C. Toole, Jr., C. Freeman, B. Cassady, et al. Cartouche: conventions for tangibles bridging diverse interactive systems. In *Proc. of TEI*, pp. 93–100. ACM, New York, NY, USA, 2010.

[18] B. Ullmer, H. Ishii, and R. J. K. Jacob. Tangible Query Interfaces: Physically Constrained Tokens for Manipulating Database Queries. In *Proc. of Interact*, pp. 279–286. IOS Press, Netherlands, 2003.

[19] D. Young and B. Shneiderman. A Graphical Filter/Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation. *Journal of the American Society for Information Science and Technology*, 44:pp. 327–339, 1993.