

Papyr: Automatically Generating “Smart” Paper Tools for Low-resource Settings

ABSTRACT

Despite the recent push toward leveraging information and communication technologies (ICTs) for replacing paper-based tasks, there remain many barriers to designing deployable and appropriate technical solutions that replace paper. As a result, paper tools such as forms, charts, and graphs continue to be widely used in low-resource organizations working in developing regions. Unfortunately, existing paper tools are often designed to be out of reach for end-users and depend on intermediaries to input data and interpret information. We apply ideas from computability and human-computer interaction to design a system to automatically generate paper tools that provide immediate computation, visual feedback, and independence from both ICTs and intermediaries at the point of use. **Papyr** is a system that is easy to use and allows low-resource organizations to quickly transform data tracking or information dissemination requirements into printable “smart” paper tools. In this paper, we explore how paper tools are currently being used by organizations in Ghana, explore the design space for smart paper tools, and describe **Papyr**’s design and implementation.

1. INTRODUCTION

Even as computing devices proliferate, often with the singularly ambitious goal of eliminating paper from a given ecosystem, the fact that paper continues to be pervasive in professional and personal spaces is undeniable. Paper is light, low-cost, familiar, accessible, easy to use, intuitive to manipulate, and convenient to distribute [18, 19, 9]. Moreover, in low-resource settings where technology infrastructure, such as stable electricity, cutting-edge computer hardware and software, and skilled human capital, is deficient, paper is an obvious choice because it can accomplish many of the same tasks that computers can, including tracking, recording, and look-up. Furthermore, generations of paper-based organizational and end-user tasks have ensured that national regulations, at least in most of the developing world, seek compliance with paper-based record keeping rules. The

enduring legacy of paper, especially amongst current paper-based workflows, provides inertia against switching to more technically intensive replacements. It is little wonder then that paper continues to be the information medium of choice in many low-resource organizations providing community health and microfinance services in the developing world.

Still, paper is not without its constraints. The most pertinent drawbacks of paper when compared to information and communication technology (ICT) alternatives is its limited ability to manipulate data once it has been inscribed, lack of “computational output”, and difficulty in digitization. Electronic media that is designed and built to replace paper possesses an almost inverse profile of strengths, but, in being unable to capitalize on paper’s more accessible and tangible properties, also has opposing constraints [9]. Can paper itself be designed to overcome some of these limitations? Can paper, for instance, make its static content dynamic? Or, perhaps, even more ambitiously, can paper provide computational feedback to its end users?

There already exist many examples of what we call “smart paper” that demonstrate how these lofty objectives are in fact possible. Paper graphs, for instance, can provide computational feedback. One example is the nomograph, a graphical calculating device that has three scales to represent a three-variable equation. By plotting along these scales using just a pen, one can determine their ideal body mass, to give one illustration, by locating their results within the “overweight”, “desirable”, or “underweight” categories, while accounting for any gender differences [20]. The partograph, another example, is a graphical tool that can provide predictive feedback to doctors or midwives throughout the progress of labour [11]. The partograph, in fact, is an especially interesting example of “smart paper” given its popularity in low-resource settings [7, 21]. It is a paper tool that is cost-effective and accessible, a tool that when coupled with well-defined protocols and appropriate instruction has been shown to demonstrate distinct benefits in the developing world [6, 13].

These examples of “smart” paper tools indicate a particularly compelling research problem of leveraging paper’s many useful properties and developing more standalone tools, while simultaneously validating their usefulness in low-resource settings where paper is cheaper, more familiar, and more accessible than its technology counterparts. Our study addresses this research problem with the following contributions:

- Needs assessment of low-resource settings for microfinance and health in Ghana through an appraisal of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DEV’14 San Jose, California USA

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

existing paper artifacts and understanding of information gaps. (Section 3).

- Development of “smart” paper tools that are pushing the boundaries of paper and exploring how paper can contribute to existing processes, especially with low-literate populations in mind (Section 3).
- Discovery of basic design principles that highlight the constraints of paper and user interaction with “smart” paper tools (Section 4).
- Construction of Papyr, a system for intermediaries from low-resource organizations to automatically generate “smart” paper tools by simply defining their tasks and their associated parameters. The intermediary can then select from suggestions based on resources available and capabilities of the end-user to print and construct the paper tools based on the resources available (Section 5).

We believe Papyr is an unusual but appropriate marriage between paper and technology for developing contexts: a computer is used to support the organization’s design of context-specific paper tools, but the produced paper tools require no ICT support or leverage at the end user.

2. RELATED WORK

The oldest vision for the integration of paper and computers may perhaps be the conceptualization of Digital Desk [22]. Digital Desk sought to more seamlessly integrate paper and digital documents by coupling paper with graphics projections, enabling interactions with paper in a way that allows for the completion of typical computing tasks. Still, the most recognizable paper-computer relationship is manifest in the printer itself [16]. However, when it comes to existing work that is directly related to this study, the printer is mostly used as a means to generate interesting paper artifacts that is an integration of computation and paper. For instance, the printer has been leveraged to generate 3-D shapes or pop-up sculptures [1, 8]. Similarly, for this study, we rely on the printer to generate standalone, smart paper tools that provide augmented computational and visual feedback.

The emerging field of “paper computing” is a step in the direction of more seamlessly integrating paper’s inherent, constructive affordances with the computational power of technology, and in doing so, rejecting the very notion of a “paperless world” [10]. Current work includes augmented paper documents, where paper sheets are digitally enhanced through the use of printed visual markers such as QR codes [10] or through the inscription of digital notes via specialized devices [2, 14]. Still other applications include leveraging paper itself as an interface to computer applications; these interfaces may be paper-based [3, 15] or paper-like [16, 5]. In addition, there are various examples of paper computing tools that have been designed specifically for the developing world. For example, CAM, a user interface toolkit, uses a mobile phone with a camera to record and store data from paper logs in rural microfinance groups in India [12]. Digital Slate, an example of a paper-like interface, is a low-cost electronic slate that allows for pen-to-paper writing motion to capture data in savings groups in India [17]. Shreddr is another example of paper to electronic data conversion technology that can be used in any low-resource organization that remains heavily dependent on paper [4]. However, these paper computing tools that have been designed for the

developing world almost exclusively focus on capturing and converting data, while ignoring a range of other tasks.

In general, in the best-case scenarios for paper computing tools, the constructive benefits of paper and technology are leveraged equally. In the worst-case scenarios, as is usually the case, these instances of the integration of paper and technology almost always end up compromising on paper’s low-cost and easy availability, or underutilizing its computational power. In contrast, this project focuses instead almost exclusively on exploiting and preserving the valuable affordances of paper by bringing in technology to furnish the *supporting* infrastructure to generate smart paper tools. This unique stance is justified in low-resource settings where paper is inexpensive, easily accessible, and familiar, thereby mitigating resource, training, and infrastructure costs [4].

3. EXPLORING PAPER

For this project we focus on specific problems within the health and microfinance contexts. After establishing the problem space, reviewing the related work, and identifying real-world problems in low-resource settings where paper is already deeply rooted and in constant use, we held several brainstorming sessions to ambitiously explore the range of computational and visual feedback that paper in and of itself could provide. We then explored how paper was being used in the health and microfinance contexts at field sites in Ghana to assess how paper is currently being used. Finally, we designed several improvements to existing paper tools and iterated on our designs with real end users.

3.1 What Can Paper Do?

The primary goal of these preliminary brainstorming sessions was to push the boundaries of what “smart paper” could accomplish. In other words, how could we re-design paper in a way that makes it somehow better? Here, we describe a small subset of the “smart-paper” tools we designed in our brainstorming sessions.

3.1.1 The Subtraction Band

The subtraction band gives the difference of numbers. The base is made of stiff paper, plastic, cardboard, or other material that could support two narrow slits through which a closed paper loop could be smoothly slid and rotated. Figure 1 is a representation of this tool where the numbered upper region is the base in reference to which the paper loop, or darker region, would move.

To take a simple example, we shaded ten boxes on the number line on the base. We now want to subtract “3” from “10” (Figure 1a). We locate “3” on the number line and remove the stickers until that point. After this slide the paper band to the left such that the first shaded cell moves back to “1”. We can now see that the result is “7” (Figure 1b). Alternatively, if materials are available, shading cells could be replaced with sticking stickers.

3.1.2 The Addition By Matching Tool

The addition by matching tool calculates the sum as well as compares and matches analogous variables. Again, the base would be made of stiff paper, plastic, cardboard, or other material that would support two narrow slits through which a paper strip could be smoothly slid up or down. Figure 2 is a representation of this tool, where the axes indicate

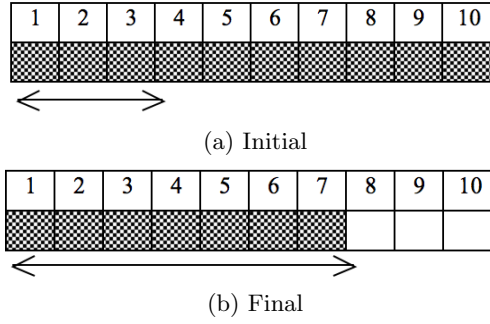


Figure 1: Subtraction band initial and final state.

the base in which each separate strip, or column, would move in relation to.

Income inflows for low-income populations typically engaged in the informal labour market are irregular and variable, accumulated through a variety of financial instruments such as money earned through different jobs and money borrowed through different sources of credit. This tool can be used to compare income and expenses, for instance, and determine when income-expenses can be met or matched or when one exceeds the other. To do this, we first shade in the requisite number of 100 cedi¹ cells to fill up the projected income and expenses column on “Day 1”. We then fill in cells in the income and expenses column on “Day 2”, starting from the point where income and expenses leveled out, respectively, on the previous day. This is repeated for every successive day. At this point, the tool is able to tell the user when expenses exceed income.

In this example, user estimates that the incomes for the first day will be around 200 cedi, shading cells to fill that amount. The user then estimates that on the second day, income will be approximately 300 cedi. He or she would then fill the cells from the bottom up to the 300 cedi point and then shift the band so that the starting point of the shaded cells is at the point in which they left off, or at the 200 cedi mark. The resulting configuration can be seen in Figure 2a. The user is now able to tell that over the two days, the income has accumulated to 500 cedis. And after populating the expenses columns in a similar manner on the tool, the user can see that expenses exceed income on “Day 1”, income exceeds expenses on “Day 2”, and they match on “Day 3”. The tool also indicates how much money has accumulated at any given time, by checking the scale on the left.

While tracking income-expenses, if there is an unexpected expense unit that is introduced on “Day 1” (represented by the red cell in Figure 2b), then the successive stacks of expenses must be readjusted. This is achieved by sliding the paper strips representing expenses on “Day 2” and “Day 3” one unit up. The tool is therefore able to readjust to match the dynamic nature of money flow for low-income populations.

3.1.3 Averages Tool

We also developed a tool that could help estimate averages. Averages, for instance, can be particularly useful in

¹local currency corresponding to roughly 0.28 USD per cedi at the time of writing

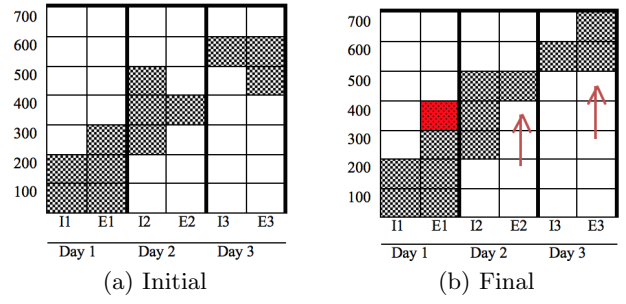


Figure 2: Addition and matching tool initial and final state.

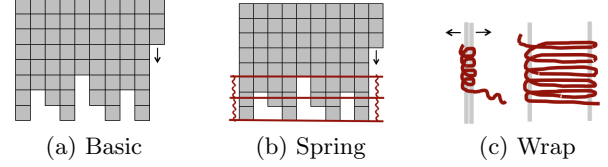


Figure 3: Different iterations of a tool to find averages.

calculating the length of a typical menstrual cycle in order to predict the length of future cycles. In its simplest form, a rolled strip of paper can be incrementally unrolled by a standard unit size to track each day the individual is not menstruating and then ripped or torn when menstruation occurs.

When these strips of paper are layered up on top of each other, the thickest part of the strips would indicate where the average would be located. The next time the individual is unrolling the strips of paper, they can know where to anticipate their cycle to end. Additionally, the halfway point is a good approximation of when ovulation would occur. A similar process could be done with the strips of paper folded in half to have a rough estimate of when the individual is most fertile.

Arranging these strips of paper next to each other would also give a good “eyeball” estimate of the average, as seen in Figure 3a. This could be enhanced with the use of a tool that lines up with shortest strip and the longest strip in the range of data points and, with springs or rubber bands attached to the middle mark, draws a line between the two to come up with an alternate means of estimating the average as seen in Figure 3b. These two do not necessarily give a very accurate average, but each has its own strengths.

Another potential design is taking thinner but sturdier strips of paper or string marked with standardized units. The strip is unrolled during the current cycle from the end of the last cycle and all of it is wound around two thin stiff rods, such as a pencil, the number of times that is the number of cycles. These can then be readjusted so that the strip comfortably spans the two rods in a way that results in equivalent sections for the number of cycles. A graphical depiction is seen in Figure 3c. This can give a fairly accurate estimation of average. However, there is a tradeoff because this is one of the prototypes that is the hardest to implement due to the need for very specific materials and complicated gestures.

3.2 In the Field Design Iterations

After our ideation sessions, we developed context-specific

tools that were designed specifically to address any deficiencies in existing paper artifacts we observed. We conducted contextual inquiries in the domains of microfinance and public health in Accra, Tema, Dodowa, and Awutu-Senya in Ghana over a period of three weeks in July 2014. During this time, we observed the organizational settings, user practices, and existing paper artifacts. We briefly describe these artifacts, the tools we designed in response, and their testing and subsequent iterations during our time in the field.

3.2.1 Susu Tracker

We spent time with a microfinance institution (MFI) in Tema, observing their daily “susu” (savings) operations. The susu process itself is straightforward. Clients deposit a standard susu deposit amount for 31 days, of which 30 days worth of accumulated sum is theirs to withdraw at any time and 1 days worth of deposit is the commission for the MFI. Therefore, the deposit that the customer made on the first day of a new cycle is immediately forfeited to the MFI as their commission. A standard paper passbook was used to record these transactions. The passbook provides the date on which a deposit was made, the amount deposited, and the running total up until 31 days. Therefore, if a customer had saved over multiple 31-day cycles without making any withdrawals, he or she would now have to add the accumulated sum per completed cycle, which would be a standard amount, plus any arbitrary amount for a partially completed cycle. Moreover, the passbook provides no commission information. More specifically, the running total for one cycle did not indicate to the user in any way that one day’s worth of deposit would be immediately subtracted from the accumulated sum, should they choose to withdraw that amount.

After observing these deficiencies, we designed a tool that could give the customer this missing information. Most of the existing customers had low-literacy and arithmetic skills, so we decided to keep the input process as simple and straightforward as possible. One of the first iterations used a calendar layout along with a stack of stickers that had the running total and commission information denoted by a red sticker that accumulated in arithmetic progression every 31 days (Figure 4a). After five design and usability testing iterations, our final susu tracker represented a simple passbook format (without stickers) where the dates and the running total were already printed out (retaining the commission information as before), and customers only had to mark or tick off the days on which they made a deposit using a pen or pencil (Figure 4b).

3.2.2 Graph Reader

At a government district and a children’s hospital in the greater Accra region, we observed daily medical practices and conducted interviews with different doctors and nurses. We also interviewed medical professors and intermediaries, such as local community health volunteers. Through our interactions we learned that women are required to maintain two booklets before, during, and after pregnancy. One booklet was used to track the health of the mother, and the other, of the child. Unfortunately, the information contained in these booklets was presented in a very complicated manner, primarily to be used by trained medical personnel.

In the booklets there was a chart that mapped the age of the child over the weight of the child to determine whether the child is at a healthy weight-age balance based on the

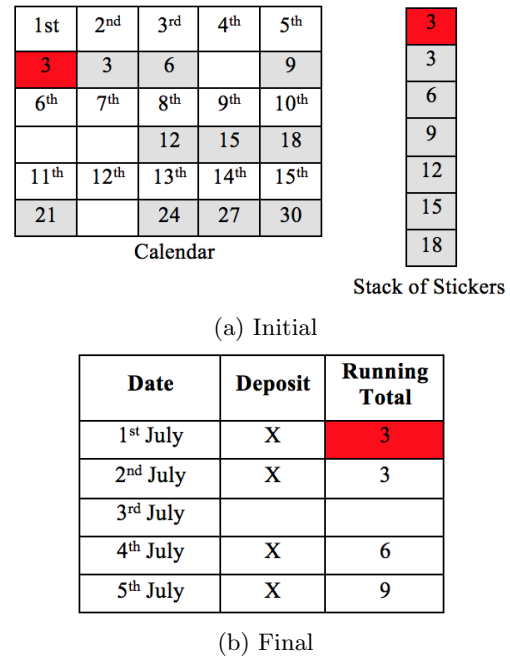


Figure 4: Susu tracker initial and final state.

WHO regulations of child growth percentiles. This chart also tracks the child’s weight over 24 months, accompanied by a legend to indicate the possible slopes between each of the points, and their implications (that is, points with a negative slope, or where the second dot is lower than the first, indicate that the child is losing weight and is in a danger zone). An adaptation of this legend can be seen in Figure 6a. Through the interviews, we discovered that this information would in fact be relevant to the mothers. But given the confusing and esoteric presentation, it was inaccessible for those who did not know how to read graphs, and clunky to use for those who could read graphs. Therefore, we decided to develop a tool that would help individuals read this graphical information more easily.

To improve this tool for low-literate populations, the first challenge was finding a way pinpoint the location of the information of interest. Various ideas for tools were developed, including tabs that stretch to the numbers of interest on the graph and position the basic square viewfinder as seen in Figure 5a. Another idea was hollow strips that could be placed on the axes to point to a given region and would overlap to pinpoint the specific region in the center, as seen in Figure 5b. However, based on design principles for low-literacy users and feedback from the needs assessment, we decided we not only needed to highlight the important regions but also to cover the information that is not needed. Therefore, we toyed around with overlapping and sliding pieces of paper and developed a tool that uses horizontal and vertical tabs to slide the region of emphasis in place. An example of this tool can be seen later in the paper in Figure 11b.

Another design question was finding ways to compare two points with each other. This could involve something as basic as the slope comparison that the current child and maternal health records use. Or, this could use the viewfinder’s space to compare the slope of two points. After feedback

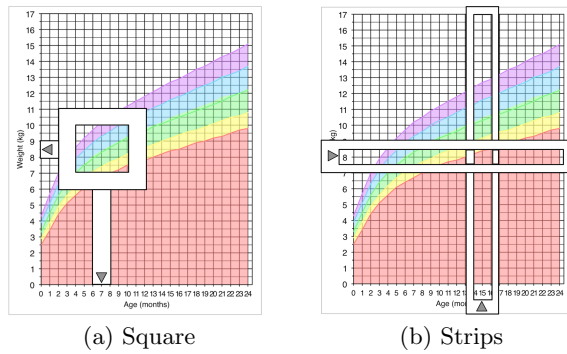
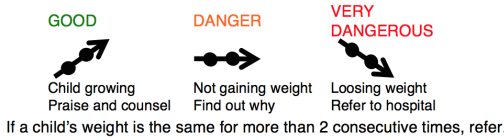
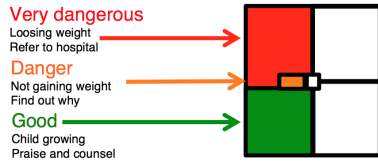


Figure 5: Different ideas for narrowing focus on a graph.



(a) Baseline slope



(b) Smart viewfinder

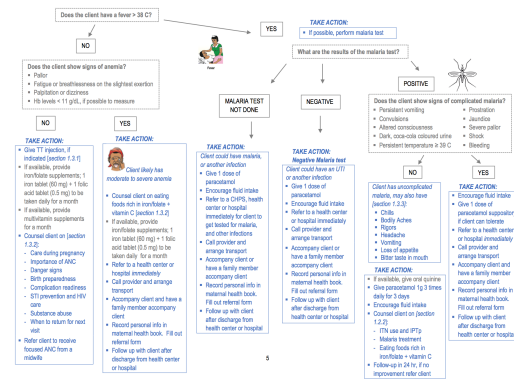
Figure 6: Different ways of comparing two points to determine change over time.

from nurses and doctors, the final iteration of the tool includes a viewfinder that compares points based on the quadrants they fall in, as seen in Figure 6b.

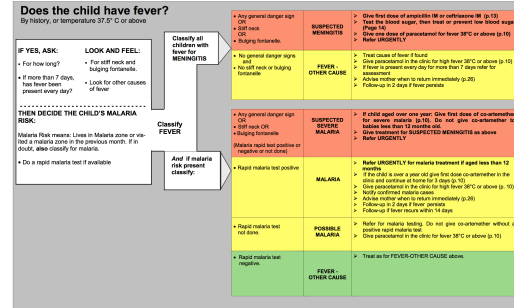
3.2.3 Flowchart Booklet

Through interviews with a reputable nonprofit organization engaged in public health interventions and interactions with community health volunteers, we learned that one of the tools used by nurses is a decision tree for diagnosing common ailments. This is derived from the Integrated Management of Childhood Illnesses (IMCI), which is a flowchart that instructs the intermediary which questions to ask the caretakers of the patient or which symptoms to check for and how to proceed depending on the results of these inquiries. However, the current representation of the logic for this process is too overwhelming with a lot of detailed information on a single diagram as seen in Figure 7a or oversimplifications to accommodate information in a way that is easily represented by a single diagram as seen in Figure 7b. Therefore, we decided to create a more efficient tool for presenting this information.

We started by simplifying the flowchart so that each potential option would be represented by a different section of the booklet. However, we soon found that this would decrease the amount of space that we would have on the page and detract from our ability to properly show all of the information. So we decided to use tabs to demonstrate the options that a user could take at a given point. After demonstrating our first iterations to an audience of doctors and nurses, it was found that the progress from one step



(a) Full flowchart



(b) Simplified IMCI flowchart

Figure 7: Different flowchart examples.

to another was not self-explanatory. Even though the users would follow the steps on the booklet, it was difficult to know when to stop. We tried adding arrows to the pages to indicate the flow from one page to the next, but the arrows were unclear when it came to needing to choose which option to flip. Finally, we found that we would make the options more clear but also to append a long red bar that said “STOP” at the end of a given action sequence. Since any “smart” paper tool in a low-resource setting will most likely not come with many user instructions or training sessions, the progression from one step to another has to be fairly clear to the user. The final design for this tool can be seen later in the paper in Figure 13b.

4. DESIGN

Given the immense design space of paper-based tools we necessarily had to limit our scope for this project. We reflected upon the paper tools that we encountered and explored in order to abstract our ideas into basic primitives that could be shared across tools. We then considered the various constraints of paper itself, our target users, and our specific contexts to scope the eventual design of Papyr.

4.1 Paper's Limitations

While the goal of this project was to manipulate paper's many properties in a way that could augment its computational and visual outputs, we were constantly aware of its limitations:

- Paper has **limited computational feedback** when compared to a computer. As an example, while generating the running total in the susu tracker, we were only able to do so because the intervals were of a fixed

value. Therefore, providing a dynamic running total for successive numbers with a variable interval between them would be challenging.

- Paper is also constrained by the **limited amount of physical space** that is available for a specific tool. For instance, the range and the upper limit of numbers in the subtraction band must be directly correlated to the size of the base paper/cardboard sheet the user is willing to work with, and that is not unwieldy and difficult to manage.
- Paper has **limitations on physical manipulation**. Paper sheets, of course, can be folded or placed next to each other, in order to provide some dynamic re-sizing capabilities, but physical manipulation such as folding also has its limits with respect to the number of folds possible with paper. Paper can provide sliding or pulling/pushing movements (e.g. in the subtraction band as well as the addition/matching tool), but such movements can be awkward or suffer from wear and tear after repeated use. Comparable, low-cost material substitutes, such as ribbons, may improve certain problems, but we limit our investigation here to paper.

4.2 User Constraints

There were constraints on the tool themselves in terms of how self-explanatory or intuitive they were to the end-user, especially to the low-literate populations that we were targeting. User capabilities can be improved through training and other in-person support, but we limited our investigations to paper tools that required minimal training and intermediation.

During our time in the field and through our interactions with the different users, we also discovered that their goals did not always match the broad goals that we had envisioned for the paper tools. This is not necessarily a user constraint per se, but it did certainly place some checks on the tools we designed and tested in Ghana. For instance, we assumed users wanted more complete and precise results that the susu tracker was providing them. However, in many cases, users were actually satisfied with the current pass-books where they were able to approximate their running total, albeit with some amount of effort. This points to a general inertia towards the learning and implementation of new tools, paper or otherwise. Still, retaining the original look-and-feel of an existing paper tool, something that is more reasonable to accomplish with a new paper tool, like we did with the final iteration of the susu tracker, can help alleviate some of this inertia.

Some of our more sophisticated tools involved too much effort to use, which did not suitably justify the benefit that users were able to glean from them. For instance, the effort invested in using the subtraction band tool may be disproportionate to the benefit that users receive from the tool, especially when their goal is not precision, but is instead a suitably accurate approximation. For similar reasons, while we discussed many paper tool options that would make use of such actions as ripping paper to show approximations, layering paper to demonstrate an increase in magnitude through thickness, or even origami to solve more complex problems, we did not end up designing or iterating on many of these ideas. We limited our Papyr prototype to support only the subset of primitives that we felt confident could be of practical use for health and microfinance problems we encoun-

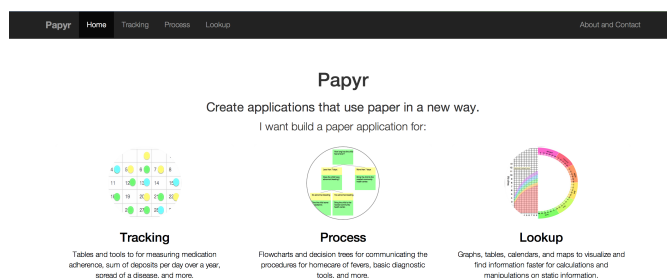


Figure 8: Screenshot of Papyr interface.

tered.

5. PAPYR

Papyr is an interactive system that allows intermediaries working at low-resource organizations to rapidly and automatically generate customized paper tools that give computational and visual feedback. Through a series of questions, the intermediary, as the user of Papyr, specifies a task or tasks. The user is then presented with a set of potential tools generated by Papyr and selects the tool that best suits the resources available and the capabilities of the end user.

In developing Papyr, our main goal is to link the paper primitives that we extracted into an actual usable system. Considering all of the potential types of tasks, the primitives paper could take on, and tools the system could output, we define three different categories of tasks: tracking, process streamlining, and information lookup. The main distinction is whether the information presented is dynamic. Tracking tasks are those that require user input at given time intervals and the resulting information, such as the sum or regularity of a process, is dynamic. Information lookup tasks are ones where the information is static and minimal user interaction is necessary to discover the information. And finally, process tasks are those that require user input to communicate the current, dynamic situation and determine which information would be shown, but the information itself is static.

5.1 Tracking

Tracking encompasses tasks that require periodic user input. This includes tracking the days a patient takes his or her pills over a month to visualize the regularity of medication consumption. Alternatively, it also includes tracking the days the microfinance savings customer makes the required two cedi deposits to calculate cumulative sum over half a year. Other use cases include caloric intake each day over a month, amount of sleep each day, length of menstrual cycle each month, mood each day, journaling of events that occur each day, or spread of disease throughout a community over a month.

These tasks all have an element of time and the information achieved relies on this element of time. In the most basic form, each of these tasks can be done using a table. The main dimension of this table reflects the change in time, with each row as a single time interval. This varies for each task based on the frequency and the duration for which the task is tracked. Another dimension is the values in each one of the cells. Some tasks require only a simple yes or no of whether the event occurred to determine regularity, such as

medication consumption, and other tasks require a numeric value that could contribute to a cumulative sum, such as daily deposits.

In the cases where there is more than one piece of data being tracked over time, each task can be defined separately and then, if the time intervals are compatible, automatically merged by our system into a single table with multiple columns. For example, if the task is to track mood every day, and there are five different possible moods, each of the moods is considered a separate tracking task and merged into a single table with one column for each mood. Users can click a checkbox to disable automatic merging of tasks if this is not desired.

The next challenge is allowing users to input task specifications and parameters in an understandable way and to translate this input to paper tool outputs. We originally proposed to allow the user to string together a series of “per” and “over” phrases for each relevant time and location to specify their task. For example, an input could read “I am tracking the regularity of my medication consumption per day over three months.” However, this method proves to be fairly ambiguous and difficult for both the user and the computer with more complicated inputs. Eventually, the minimal series of questions was developed to determine the dimensions of the tracking task and differentiate between each of the potential outputs.

The resulting system asks the user a series of questions about the tracking task they are proposing, including the values of the input and the frequency and duration of the tracking. The system is interactive—for each question, Papyr iterates on the user’s answers and outputs suggestions, allowing the user to visualize the impact of each additional constraint. For example, after answering the first question, “What are you tracking?”, the user is presented a blank one-column table where only the first row is filled with the task name. The next question asks about the possible values of the tracking task. If the user indicates that they are tracking only whether the event occurs or not, another suggested output is the same one-column table populated with “YES/NO” in each row so that the end-user can circle the corresponding word based on the occurrence of the event. The user can very clearly see the incremental difference between the blank table and the new “YES/NO” table when choosing between the two.

Each of the output suggestions includes the base table that will be used, as well as a set of possible instructions that vary based on the level of literacy of the end user and the cost of any external materials. In the previous example, for the blank table, the user receives potential instructions of writing either “Y” or “N” on each of the lines depending on whether the event occurs, shading in the rows for the time intervals the event occurs, or placing stickers in those rows.

The user then has the opportunity to compare all of the possible tools along with their associated instructions and select the one that matches their specific context. A screenshot of the Papyr Tracking workflow is seen in Figure 9. The potential table-based outputs are similar to Figure 2a and Figure 1b for “sticker with calendar” and “no sticker with merged tables” outputs respectively.

5.2 Information Lookup

While tracking deals with tasks that require user manip-

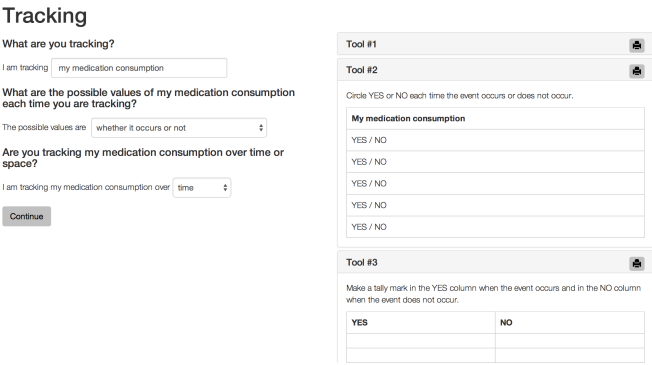


Figure 9: Screenshot of tracking interface.

ulation and input, other tasks deal with static information. These include tasks such as calculating an individual’s body mass index (BMI) to looking up whether that BMI falls in the percentiles for being over or underweight. Another would be finding out the number of calories for each food item eaten or checking to see if a child’s age and weight are at a healthy balance. The most common forms of static input and output are calculations and information lookups. And in reality, calculations can be construed as information lookups because the answers to a calculation could be enumerated and placed in a lookup table. As with the tracking baseline representation, our baseline representation for lookup is in the form of tables.

The dimensions of these tables are essentially the inputs (i.e. a lookup key) and the outputs (i.e. information the end user is looking up), which are likely numeric or can be reduced to being numerical i.e. categorical). In a basic lookup table, all of the inputs would be in the first columns and the associated outputs would be in the columns that follow, with all possible combinations in each of the rows. This basic layout can be seen in Figure 10a.

However, there are many cases in which there are more appropriate ways to represent this data than this large, cumbersome table. In a more specific example, where there are two inputs and one output, the inputs and outputs could be arranged such that one input is the columns and the other is the rows and the outputs as the corresponding intersecting cells. This can be visualized in Figure 10b. In another case where there are potentially three inputs and one output, a similar row-column structure would work for the presentation of two of the three inputs and the final input would be factored into the separation of the table into tables based on different values for that input. This can be visualized in Figure 10c. And with even larger numbers of inputs, the split could be across different pages, with different colors, and other methods that engage the dimensionality of paper primitives as discussed in the abstraction section.

For low-dimensional lookups, tables are relatively easy to use. For higher-dimensional lookup tasks, alternative visualizations are beneficial. A clock or calendar could be used to visualize information that has to do with time or date. Additionally, information could also be presented on a map. Due to lack of space, the two specialized lookup outputs we present here are graph representations and calendar year calculations.

Graph representations cover data that has two numerical

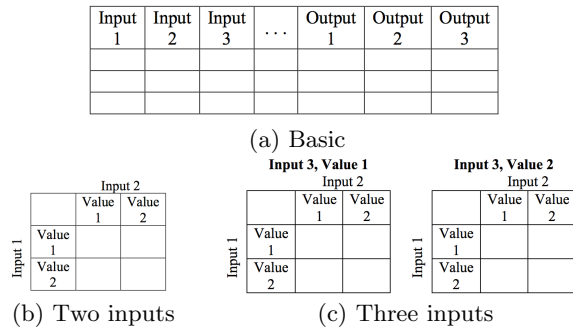


Figure 10: Different arrangements of lookup tables.

dimensions as inputs and a numerical or categorical output. For example, if someone knows their height and weight as numerical inputs, they can figure out whether they fall in the category of under, normal, or overweight based on a graph with those categories as differently colored regions.

Paprr suggests graph enhancements to help end users with understanding the content of the chart or encourage interactivity. These improvements are drawn from the ideas and feedback collected from the graph reading tool we proposed. The first improvement includes tabs and cutouts to help narrow the end user’s focus on the relevant region of the graph. For graphs with two axes, this is a vertical sliding cover to isolate based on the y-axis and a horizontal sliding tab to isolate the x-axis. The user can elect to use these components together or separately. This can be seen in Figure 11 which shows the output of a graph tool that has the sliding tabs that pinpoint the relevant information to help block out the rest of the graph.

A second enhancement is a “smart viewfinder” that guides the end user in doing some basic approximations. For example, sectioning the “smart viewfinder” into different quadrants in relation to the midpoint helps to determine if the slope of the line from the previous point is positive or negative (e.g. Figure 6b). A notched “smart viewfinder” could help the end-user compare the distance between the two points, whether it is comparing the sizes of bars on a bar graph or the amount needed to increase or decrease to be in a different category or region.

In order to build these graph-based information lookup tools, users will be asked to input the data that they want to work with, and a graph will be built. Then, users will be presented with the pros and cons of each graph improvement and asked which enhancements they want, whether it is the vertical or horizontal tabs or the viewfinder.

Calendar year calculations are a specific type of calculation for which we developed a paper tool. This tool is made of two concentric circles, the smaller layered on top of the larger. Users can specify the relative number of days between one event and another, provided the difference is within the 365 day range of a year. Then, by spinning the inner circle to line up with a date on the outer ring, the end user can calculate the date of the next event. For example, gestation occurs approximately 280 days after the last menses. A tool for this aligns the 0 day mark with the date of the last menses, the 14 day mark with conception, and the 280 day mark lines up with the predicted gestation day. This sample tool is seen in Figure 12. To generate this tool for other calculations, the Paprr system takes as an input an

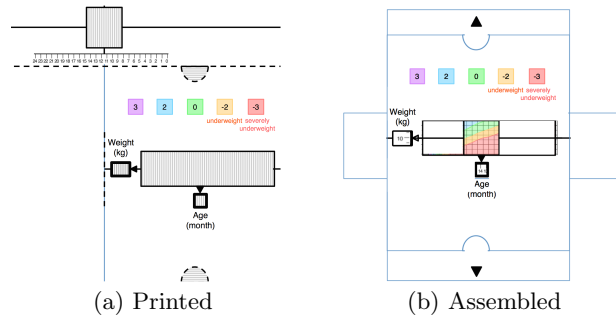


Figure 11: Output of the graph generation tool printed and assembled.

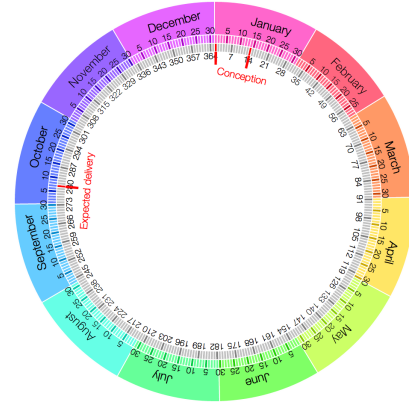


Figure 12: Example of calendar year calculation tool.

indication of which days are matched with which important event in the year range.

5.3 Process Streamlining

A specific case of information lookup is process driven—though it has static information, the relevance of this information depends on the specifics of the situation. This information could be presented using a table, with each of the potential factors that contributes to the classification of a given situation as an input and the resulting instructions as an output. However, this would result in a fairly sparse table or many sets of sparse tables separated on different pages or sections. More commonly, this type of information is presented in a flowchart or decision tree. The tool produced by the process module of Paprr can be used to depict this logic, including, but not limited to, questions community health volunteers could use to diagnose a patient, decision diagrams to help a customer choose a savings scheme, or checklists for procedures that change based on the outcome of a step.

In the Paprr system, the user can specify the logic of the process in flowchart form. The flowchart is a directed, acyclic graph whose nodes are the “steps” or instructions the end user should follow and whose edges are “options” or logical connections from one step to another.

For example, if the user is looking to create a tool to help parents with the home treatment of fever in their children, the user would first create a step with the question of “How long has the child had a fever?” From this question, there would be two possible paths that can be taken. The user would create another step connected to the first step

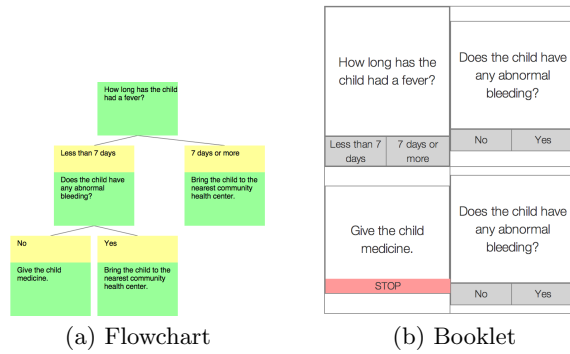


Figure 13: Process logic depicted as a flowchart and as a booklet.

with the option “less than seven days” and the instruction to “bring the child to the nearest community health center.” For the other possible answer to the question, the user would create another step that is connected to the first question with the option “more than seven days” and with the question “Does the child have any abnormal bleeding?”. The user would then continue this process in specifying the answers to this question and continue until no questions remain and all paths end in instruction. This example is presented in Figure 13a.

Then, the system converts this logic into a booklet. Each page of the booklet will present one question from the flowchart and the potential answer options associated with it as tabs below. The end-user can flip the tab over of the option that they select and continue with either the question or the instruction on the page. An example of the output of the Papyr system is show in Figure 13b.

Some of the challenges for this workflow in Papyr include figuring out the order of the steps and instructions. In order to incorporate the feedback from the needs assessment that there should be no blank pages, balanced with the need to show visible instructions, whether it is all possible options at a given time or a “STOP” instruction, our resulting algorithm for adapting the process logic from a flowchart to a booklet ended up being a depth first traversal that highlighted all nodes, or steps, that were accessed, even those that had been visited before. At each node, the algorithm would access the children of the node and append these options as tabs at the bottom of the page but if there were no children, append a “STOP”. Another challenge was determining the best way to fit all of the instructions on a page, as per the “screen real estate” constraint of paper. The default presentation of information is using a fourth of an A4 sheet of paper. However, since there seemed to be varying preferences from the feedback, the user also has the option to adjust the size of each of the pages as well and our system uses this to parameterize the output.

5.4 Implementation

The system is implemented as a web-based system with HTML/CSS and Javascript built utilizing the jQuery, Twitter Bootstrap, and D3 frameworks in 2134 lines of code.

6. DISCUSSION

The motivation driving this study is the fact that paper is abundant, affordable, deeply embedded in existing work-

flow practices and personal spaces, and familiar, especially when compared to digital devices that are yet to achieve the same level of adoption and use in the developing world. Furthermore, as we demonstrate in this work, paper is capable of providing augmented computational and visual feedback to achieve some of the everyday tasks that computers currently do. Smart paper can perform basic arithmetic tasks, provide regularity and frequency information through visual feedback, and streamline information dissemination.

In general, when aiming to solve a problem in a platform-agnostic way (should I use paper or technology?), the design motivations, principles, and applications remain broadly consistent (i.e. without considering technology, all the other design considerations remain the same). Therefore, while working within a given scope, that is the degree of computational or visual feedback required to solve a problem, we encourage a more rigorous and dynamic use of paper to resolve some of these problems, especially in low-resource settings. We believe that the fervent drive to use ICTs to solve a broad range of problems could be scaled back and reconsidered. To avoid the risk of romanticizing paper, we recognize that paper’s computational power cannot match that of a computer’s. However, paper can certainly be improved over its existing uses as we show specifically in the context of low-resource organizations in public health and microfinance.

Papyr is a supporting infrastructure that automates the creation of the paper tools. Papyr is unlike existing paper computing applications that almost always reduce paper’s presence in these applications, diluting some of its more valuable affordances, or replacing it altogether by mimicking its properties in a surrogate (and “sexy”) technology artifact. Instead, Papyr leverages technology in the design and creation processes, but the output completely preserves the affordances of paper for the end user.

7. CONCLUSIONS AND FUTURE WORK

This paper describes the an exploration into the design of “smart” paper tools in the context of developing regions. We described the process of brainstorming, designing, assessing, and developing a paper-technology infrastructure that automates the production and distribution of smart paper tools. We explored existing paper tool use in healthcare and microfinance in Ghana to understand how paper is currently being used by low-resource organizations to track and disseminate information. From our needs assessment we iterated on several potential designs of enhanced paper-based tools to push the boundaries of paper. We designed a smart paper generation tool that allows intermediaries to specify tasks to quickly and automatically construct potential paper-based solutions that do not require ICTs at the point of use. We describe the design and implementation of our “smart paper” prototype, Papyr and demonstrate how it can be used to produce smart paper outputs that preserve the affordances of paper for the end user.

There are many features that we are still incorporating into Papyr and even more areas of improvement. We believe that there is ample potential to leverage paper’s tangibility in future work. We plan to explore many of the ideas we mentioned here including: ripping, stacking, layering, or complex folding (origami) with paper. There is also potential to improve the specific modules in Papyr. For instance, future iterations of the tracking workflow will include im-

proved abstractions and optimizations from the given tables, so as to better handle more inputs across different tasks, including optimizations of using maps to track the events across geographical space. In the same vein, future versions of the information lookup workflow will produce more specialized tools including editable calendars and maps. Finally, a future optimization on process streamlining will allow the inclusion of images as components of the steps, for both the questions as well as the instructions and options to better accommodate low-literate individuals. We plan to provide **Papyr** as a free open-source platform for immediate use by low-resource organizations.

8. ACKNOWLEDGEMENTS

We thank the participants of our studies for their patience, helpfulness, and valuable feedback.

9. REFERENCES

- [1] Orihedra: Mathematical sculptures in paper. *International Journal of Computers for Mathematical Learning*, 1:225–261, 1997.
- [2] Anoto. Digital pen and paper technology. *Anoto AB*.
- [3] Q. Bonnard, R. Center, F. Kaplan, and P. Dillenbourg. A paper interface for code exploration. In *In 12th ACM International Conference on Ubiquitous Computing (No. EPFL-POSTER-158791)*, 2010.
- [4] K. Chen, A. Kannan, Y. Yano, J. Hellerstein, and T. Parikh. March 2012.
- [5] M. Coelho, L. Hall, J. Berzowska, and P. Maes. Pulp-based computing: a framework for building computers out of paper. In *In CHI’09 Extended Abstracts on Human Factors in Computing Systems*, pages 3527–3528. ACM, April 2009.
- [6] M. Fahdhy and V. Chongsuvivatwong. Evaluation of world health organization partograph implementation by midwives for maternity home birth in medan, indonesia. *Midwifery*, 21(4):301–310, 2005.
- [7] A. Fawole, K. Hunyinbo, and D. Adekanle. Knowledge and utilization of the partograph among obstetric care givers in south west nigeria. *African journal of reproductive health*, 12(1):22–29, 2009.
- [8] S. Hendrix and M. Eisenberg. Computer- assisted pop-up design for children: Computationally- enriched paper engineering. In *In Proceedings of IASTED Intl. Conf. On Computers and Advanced Technology in Education*, 2005.
- [9] W. Johnson, H. Jellinek, L. K. Jr, R. Rao, and S. Card. Bridging the paper and electronic worlds: the paper user interface. In *In Proceedings of the INTERACT’93 and CHI’93 conference on Human factors in computing systems*, pages 507–512. ACM, 1993.
- [10] F. Kaplan and P. Jermann. Papercomp 2010: first international workshop on paper computing. In *In Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct*, pages 507–510. ACM, 2010.
- [11] W. H. O. Maternal, C. Health, and F. Planning. The partograph: a managerial tool for the prevention of prolonged labour (vol. 1). 1988.
- [12] T. Parikh, S. Javid, K. Ghosh, and K. Toyama. April 2006.
- [13] K. Pettersson, M. Svensson, and K. Christensson. Evaluation of an adapted model of the world health organization partograph used by angolan midwives in a peripheral delivery unit. *Midwifery*, 16(2):82–88, 2000.
- [14] T. Pietrzak, S. Malacria, and Ñ. Lecolinet. What do u-note? an augmented note taking system for the classroom. In *In Papercomp Workshop*, 2010.
- [15] E. Portocarrero, D. Robert, S. Follmer, and M. Chung. The never ending storytelling machine a platform for creative collaboration using a sketchbook and everyday objects. In *Proc. PaperComp*, 10, 2010.
- [16] J. Qi and L. Buechley. Electronic popables: exploring paper-based computing through an interactive pop-up book. 2010.
- [17] A. Ratan, K. Toyama, S. Chakraborty, S. Ooi, K. Koenig, M. Chitnis, and M. Phiong. December 2010.
- [18] A. Sellen and R. Harper. Paper as an analytic resource for the design of new technologies. In *In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 319–326. ACM, 1997.
- [19] A. Sellen and R. Harper. *The myth of the paperless office*. MIT Press, 2002.
- [20] A. Thomas, D. McKay, and M. Cutlip. A nomograph method for assessing body weight. *The American journal of clinical nutrition*, 29(3):302–304, 1976.
- [21] A. Umezulike, H. Onah, and J. Okaro. Use of the partograph among medical personnel in enugu nigeria. *International Journal of Gynecology and Obstetrics*, 65(2):203–5, 1999.
- [22] P. Wellner. Interacting with paper on the digitaldesk. *Communications of the ACM*, pages 87–96, 1993.