The Role of User-Created Automations in Personal Health Informatics

Jacob Solomon
University of Michigan
jacobbs@umich.edu

ABSTRACT
People face significant challenges in using and managing health information effectively because the numerous systems, apps, or devices that contain personal health information are largely fragmented and disconnected from each other. End-user programmed *personal automations* that integrate disconnected systems, such as the "recipes" created on integration platforms like IFTTT.com, offer a highly user-centered solution to this problem. Through a qualitative analysis of IFTTT.com and a user study of a prototype health-focused integration platform, I show that users create personal automations that perform important roles such as consolidating interfaces, archiving information, and executing behavioral interventions. These automations expand functionality and tailor a personal health informatics infrastructure to users' needs and preferences. I also show that most automations keep users in the loop to some degree, and that there is a need to design tools for creating automations for personal health informatics that assist users in allocating tasks between the system(s) and the user.

Author Keywords
Integrations; End-User Programming; Health Informatics

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

As the number of connected devices, apps, and web systems grows and touch more and more aspects of people's daily lives, users must increasingly develop routines and strategies for interacting with their various systems and information. One area where this issue is particularly important is in health and wellness. The ecosystem of health-related information technologies is highly fragmented, requiring users to interact with many types of systems such as Electronic Health Records (EHRs), Personal Health Records (PHRs), connected devices, insurance and financial systems [27]. At the same time, people's personal health informatics is only one piece of their overall personal informatics. Health information is a part of everyday use of many systems such as email, calendars, social media, e-commerce sites, mobile phones, and many other technologies that further fragment access to and meaningful use of health information.

Under the fragmented ecosystem of health IT, users themselves are often the only link between different systems, putting a burden on users to access and interact with each system, transfer or input information from one place to another, execute actions or make decisions in response to new information. These types of tasks have high potential for automation, as they often involve structured rules, processes and information that computers are well equipped to manage.

Many systems for personal information management afford users an ability to create rules, macros, or routines to automate tasks. These *personal automations* can empower users to identify information or behavioral needs that can at least partially be performed by a system. For example, email filters that automatically move incoming mail into specified folders are a commonly used personal automation.

Recently, Integration Platforms such as IFTTT[1], Zapier[2], and Microsoft Flow[3] have been designed to enable users to create personal automations that work between otherwise disconnected systems. For example, using IFTTT.com a user with an internet-connected lawn sprinkler system can create an automation to follow a weather RSS feed and turn off sprinklers if there is rain in the forecast. Or, users can create an automation that uploads photos that they are tagged in on Facebook to a cloud storage service like Dropbox.

I argue that the integration platform model offers a promising solution to the problem of fragmentation in health IT by empowering users to create personal automations that connect functions across systems. Through an analysis of real personal automations on IFTTT.com and hypothetical automations created by users on a prototype of a health-focused integration platform, I outline several important roles that user-created automations can play in personal health informatics. In addition to predictable roles such as reducing the burden of a repetitive task or alerting a user of important information, personal automations are frequently used to consolidate interfaces to information or services, to archive information, to communicate and share information, and to execute behavioral interventions.

An important trait of personal automations is that they are end-user programmed, meaning that the creator of an automation is in most cases designing it for him or herself to use.

[1]http://ifttt.com
[2]http://zapier.com
[3]http://flow.microsoft.com

This makes an analysis of personal automations an opportunity to study the role that users assign to themselves in automated or semi-automated information management tasks. I show that users create personal automations that place widely varying responsibilities on the user him or herself, although a majority of recipes keep the user somewhere in the loop. From this analysis, I draw on Human Factors theories of human-automation interaction to argue that a critical goal for the design of tools for creating integrated personal automations for health must be to assist users in appropriately allocating tasks between the user and the systems that are part of an automation.

BACKGROUND

An ongoing challenge for the design of health information technologies has been a lack of integration and connection between the multitude of different systems, databases, and devices that collect, analyze, archive, and report health information [27]. Personal Health Records (PHRs) that integrate various systems and enable health data to be collected, stored, and delivered to both patients and providers have been promoted as a patient-centric solution to this problem [11]. Many have argued that PHRs can empower patients to exercise greater control and autonomy in their own health care [11][32][34].

Although useful, PHRs are insufficient as a stand-alone tool for managing health information [9]. While a PHR may be effective for consolidating health information, it nonetheless still becomes another disparate information source within the context of a user's more general personal information management. Pratt et al. [27] have argued that tools for the management of health information should be integrated into people's overall personal information management tools and workflows.

Jadad et al. [15] have suggested that the value of the internet to health decision making should be to provide "valid and relevant knowledge at the right time, at the right place, in the right amount and in the right format." A solitary tool for the entirety of personal health informatics will frequently fail this test as the design of a "single tool for all" will never be able to account for the heterogeneity in patients, decisions, information, and contexts that must go in to determining the right place at the right time.

Furthermore, people have health information needs beyond purely access to personal health information that is available through PHRs. The volume of health-related data that is encountered by people daily is creating conditions of information overload [1][17], creating a need to filter or curate information. Also, much of the information available to patients is intended to inform a specific action. The value of knowing that one's immunizations are outdated, for example, is to prompt a person to schedule an appointment. In many cases, the decisions or behaviors that are prompted by information can be burdensome or repetitive. Thus, the barrier to executing these decisions is not merely the availability of information but the nature of the task itself. These types of decisions and behaviors are not well supported by personal health records.

Integration Platforms

Integration platforms such as IFTTT pose a potential design and infrastructure alternative to the PHR model of personal health informatics. Rather than a single system for managing health information, integration platforms enable users to connect different platforms to automate small information tasks. Integration platforms are a merger of two important concepts, each of which has received considerable attention in the HCI literature. These platforms merge the notions of end-user programming, in which users write macros to "create, modify or extend a software artefact" [21], with web mashups, which are aggregations of content from different websites or services into a single user interface [37]. This model enables a concept often called "trigger-action programming [36], as users can create an small program that listens for a trigger event on one platform (e.g. a new file added to a cloud storage account) and then executes an action on an otherwise unconnected platform (e.g. post a message about the file on Facebook).

I argue that personal automations as enabled by an integration platform have the potential to resolve several important shortcomings of the PHR model for personal health informatics. End-user-programmed personal automations can empower users to decide what the "right amount of the information at the right place and time" are for themselves by creating automations to notify, redirect, or filter information so that they engage with it in the preferred context.

For example, keeping track of medical expenses can be a burdensome task as bills can come in from multiple providers, insurance companies, pharmacies etc. Personal automations that continuously scan multiple sources of expenses and automatically add them to a family budget spreadsheet reduce the burden of manually entering the information and also enable a user to engage with expense information in a useful context of planning an overall budget. Well-crafted personal automations therefore can help people make the most out of their health information by putting it into a useful context or filtering out information that may be misleading or irrelevant.

Civan et al. [9] conducted participatory design sessions to understand what design principles patients want in a tool for personal health information management. They found that users want such a system to afford control over information, to allow for sharing information, to enable integration of information from different sources, to allow for flexibility in the way information is captured and recorded, and to be secure. The integration platform model presents a solution that can address most of these needs.

While this model shows promise in theory, there is little evidence to date that personal automations for health can actually deliver on this promise. Current platforms such as IFTTT, Zapier, and Microsoft Flow have supported automated connections between only a few health-related apps such as fitness trackers or HIPAA-compliant encryption services. And more importantly, there is not yet concrete evidence that if users had a powerful integration platform available for creating personal automations, that they would necessarily be successful at designing automations that actually fulfill important personal health information needs, as it is not known what kinds of automations users would even design and use. A primary motivation of this work is to explore the kinds of personal automations users would create if the integration platform model were implemented in health informatics.

Personal Automations

Automations for personal informatics have been studied primarily in the context of home automations [3] and controls for the "Internet of Things" [35]. Woodruff, for example, studied how Orthodox Jewish families create and configure home automation systems to assist in observance of the Sabbath by automating activities such as turning on appliances or lights [38].

End-user programming interfaces that afford the creation of simple rules such as "if this event occurs, then do this action" provide users an effective way to connect and control different devices and apps [35][10][4]. This type of end-user "trigger-action programming" is often implemented through visual programming languages that make personal automations accessible to non-programmers [14].

There are however, several challenges that remain for designing end-user programming interfaces for personal automations. Most importantly, many users lack precise mental models that enable them to appropriately predict and understand exactly what an automation will do [14]. Thus, these automations may have good intent but fail to perform as expected, leading to frustration and errors [39][14].

Cao et al. [5], in studying end-user programming of content mashups, finds that end-user programmers must simultaneously think as programmers, designers, and users. Creating a simple rule such as those used in personal automations demands that users think about their own requirements as a user, use creativity to come up with combinations of triggers and actions that will satisfy the requirements, and think through the technical execution of the rule in constructing the automation's logic. For this reason, it is not clear how well creators of personal automations for health will perform at all three of these tasks simultaneously.

Human Factors research has frequently found that adoption and deployment of automation technologies often has unintended consequences [6], including automation bias [33], loss of vigilance [22], or loss of trust when automation fails [12]. Furthermore, in what is often called the "irony of automation" [2], adopting automation technologies frequently does not remove users from the loop, but simply changes their role in some way [26]. Often this results in users moving from a role of direct control to supervisory control [30] in which users can engage with information about system status or intervene in the system, but have little direct responsibility for controlling the system.

A persistent challenge in the design of automation technologies is deciding where in the loop users should have involvement and how to allocate functions of a task between the machine and its user(s) [20]. In other words, just how automated should a system be for a given task? This *function allocation* question is difficult for even experienced designers, and has been called more of an art than a science because most theories or heuristics are frought with inconsistencies or provide insufficient guidance [29]. However, poor function allocation in the design of automated systems is often at the root of automation failures [13].

The function allocation problem is a decision faced by user-creators of personal automations for health. What tasks or functions of personal informatics should be allocated to the system(s) within an automation, and what role should the user play? Where in the loop of an automation will users place themselves? By understanding the role users assign to themselves in personal automations for health, I hope to contribute to the design of tools and interfaces for designing and programming tools for creating personal automations that fulfill important needs of personal health informatics and empower effective use of information in health decision making and behavior.

PERSONAL AUTOMATIONS ON IFTTT.COM

To understand the role that personal automations can have in health informatics, I undertook a qualitative analysis of two different datasets. The first dataset is a collection of personal automations created on IFTTT.com. These data were collected by Ur et al. [36] and made available to the SIGCHI community for further analysis. This dataset, while not specific to health, offers an opportunity to study personal automations "in the wild" and evaluate the roles and functions that can be served by end-user programming on an integration platform for more general personal informatics. I combine this analysis with an analysis of hypothetical personal automations for health on a prototype integration platform.

IFTTT structure

Personal automations created through IFTTT.com, known as "recipes," can be created or adopted using a simple end-user programming interface. Users select a "Trigger Channel", which is an app or service that the recipe can connect to. Users then choose a "trigger", which is an event or data on the channel that the recipe should listen for. For example, the user could connect to their Facebook account and set the trigger to be any new photo uploaded by a particular friend. Users then select an "Action Channel" and the specific action they want the automation to take, such as sending the user an email about the new photo or saving it to a cloud storage account. Figure 1 shows the IFTTT interface.

The dataset contains 223,058 recipes. For each recipe, it contains the trigger channel, the trigger, the action channel, the action, a user-supplied description of the recipe, and the number of times the recipe has been adopted by other IFTTT users.

Initial Analysis of Triggers and Actions

In preparation for direct analysis of personal automations on IFTTT, I performed several iterations of open coding on the 762 unique triggers and 367 unique actions that appeared within the dataset. This initial open coding served three purposes. First, it served to bound the analysis of the created

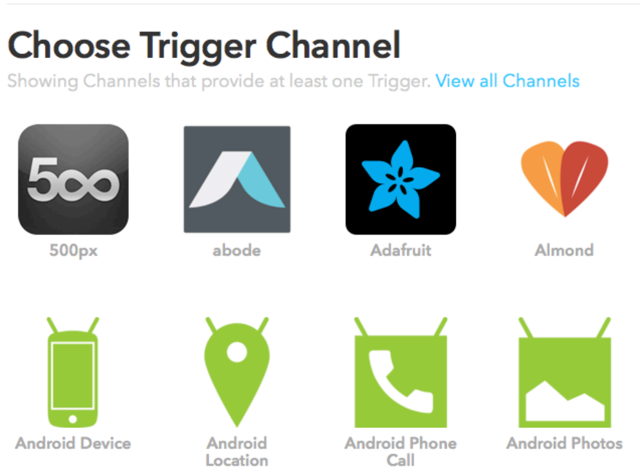

Figure 1. Interface of IFTTT

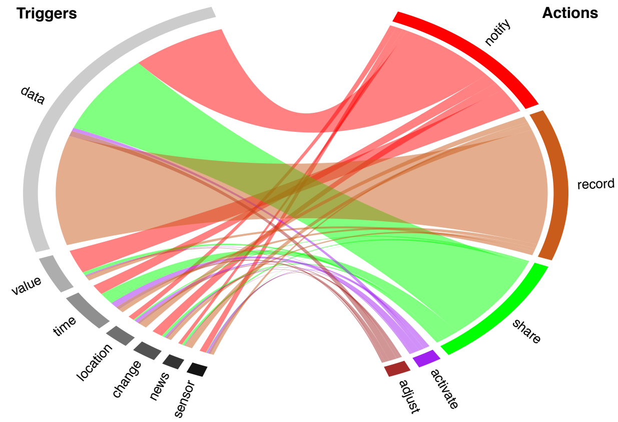

Figure 2. The number of adoptions of recipes connecting each combination of trigger type and action type on IFTTT.

recipes within the context of what types of recipes are technically possible on the platform. Second, it served as a sampling criteria for a qualitative analysis of the actual recipes described in the next section. Third, it served to inform the design of a health-oriented integration platform used in the user study.

I iterated over each list noting themes and concepts of relevance to my research questions. In each iteration I attempted to apply all previously noted concepts as codes to the item and began developing definitions for each code, adjusting codes and definitions as I went to create consistency and meaningful concepts. After four iterations through each list, I felt the applied codes gave a useful characterization of the affordances of the IFTTT platform. Table 1 lists the categories and their definitions.

Figure 2 illustrates the popularity, in terms of total adoptions, of each combination of trigger category and action category. Almost 70% of recipe adoptions use some type of data artifact as the trigger, with the majority of these being data from a social media platform or from RSS feeds. This suggests that users see personal automations as a way to monitor and filter information. Second, recipes that *activate* or *adjust* were far less popular than recipes that notify, record, or share. This is noteworthy because activating and adjusting represent more executive actions and a higher degree of automation. Most adoptions of IFTTT recipes serve to manage information by putting it in a desired place at a desired time, rather than to execute actions on their own.

Qualitative analysis of IFTTT recipes

I used the initial analysis of recipe components (the triggers and actions) to construct a small sample of recipes for in depth qualitative analysis. I took a sample of 250 recipes, with the sampling algorithm ensuring a base level of coverage of all combinations of trigger and action categories, then adding recipes from each combination based on its proportion in the overall dataset.

I developed a coding schema based off of Sheridan and Verplank's [28] taxonomy of *levels of automation*. This taxonomy identifies low levels of automation such as "Computer helps by determining the options" (level 2) and "Computer selects action and human may or may not do it" (level 4), and higher levels of automation such as "Computer does whole job and necessarily tells human what it did" (level 7) and "Computer does whole job and decides what the human should be told" (level 9). I adapted this schema to a 5-level taxonomy that fit the affordances and simplicity of IFTTT recipes. Part of this taxonomy is an assessment of the finality of the recipe, i.e. whether or not the recipe served some completed function on its own or if there was a clear subsequent action that needed to be taken, either by the user or by a system. In recipes that implied or stated (in the description) a clear followup action, I included the execution of that action as part of the overall system. For example, in a recipe that sends a notification to the user when an air filter is dirty, the act of changing the air filter is clearly implied by the recipe and therefore included as part of the overall system. I made determinations about whether or not a followup action was intended to be performed by the user or by a system as part of the determination of the recipe's level of automation.

Using this taxonomy, I classified each recipe in the sample according to its level of automation. I enlisted another researcher with previous experience coding qualitative data to provide an independent coding using this taxonomy. We met and discussed disagreements in coding and came to a resolution for the final code for each recipe.

I then went through the set of recipes within each level and followed Muller's [23] guide for grounded theory method in HCI research to identify themes and patterns that distinguish and relate different types of recipes at different levels of automation.

Findings

Level 1- The user triggers the recipe directly.

In these recipes, the user must initiate the recipe. This was most often done by the user sending an email directly to

Table 1. Categories of triggers and actions

Category	Definition	Example
Triggers		
Value	A specified value or threshold has been met	10,000 steps has been reached from a fitness tracker
Change	Something changes from one discrete state into another state	A specific product has become available at an online retailer
Time	A time has been set on a calendar or clock, or a specified time has arrived	Every year on New Years Day
Data	A data artifact such as a photo or post has been created or delivered	A new photo has been uploaded to a social media account
News	A news event has occurred	A new story about a specific topic is posted at a newspaper website
Sensor	A sensor in the physical environment has been triggered or recorded a value	A motion sensor on a security system has detected motion
Location	A location has been entered or exited	A user has left their workplace
Actions		
Notify	A directed notification is sent to someone specific	An email is sent to the user
Record	Data artifacts or metadata are recorded or logged	A new row is created in an online spreadsheet
Share	Information or data artifacts are shared in a public or non-directed way	A status message is posted to Facebook
Activate	Something is turned on or off	An oven is turned on
Adjust	A property or value is changed or updated	Lights are dimmed

IFTTT or performing some action on social media such as liking a post or uploading a photo. Some examples:

- A recipe described as “Send new photos of the kids to Mom” that sends an email whenever photos are added to a specific album on a phone.
- A recipe that automatically uploads any new photo by the user on Facebook to a Flickr account.
- A recipe described as “Help me find my phone” that places a phone call to the user’s phone when receiving an email with a specific subject line.

These recipes largely stored information or sent communications to someone. As they are triggered by the user themselves, a fundamental purpose of these level 1 recipes is to create a new user interface for interacting with a given app. In the first example, the automation serves to make the photo organization interface of the phone a new interface for sending emails. Recipes that *archive* information were the most common in this level, indicating that in spite of the availability of automation to assist in archiving information, many users maintain a desire to manually curate the information that gets archived, using the automation only to make an efficient interface for moving the data from its source to a desired archive. I should note that after these data were collected, IFTTT created a new and distinct service called a “DO button” that allows users to directly trigger automations rather than listening for events online. This service effectively creates a separate product explicitly for creating level 1 recipes where the user is designed to intentionally be the trigger for the automation.

Level 2- The user sees the outcome of a recipe action (nearly) immediately, and there is a clear followup action that must be taken by the user.

Level 2 recipes can largely be categorized as “reminders” or “alerts”. They monitor the world through a channel and notify the user when it is time to take or consider a specific action. Examples:

- A recipe that sends the user an email when a particular stock drops by 5%.
- A recipe that sends a text message on a specific interval with a reminder to drink water
- A recipe described as “Mom is coming home” that blinks the lights on a smart home system when a phone enters a specified area.

An interesting observation is that many of these recipes were connections between a *data* trigger category and a *notify* action category, where the trigger was simply a different type of notification system that presumably the user does not monitor regularly. As in level 1, this is further evidence that a primary function of personal automations is to simply put information into a more preferred interface.

Level 3- User sees the outcome of the action (nearly) immediately, but there is no clearly implied followup action.

The primary function of level 3 recipes was often to *inform* the user of something that may be important but which does not have a clear followup action. These recipes are primarily about collecting useful information and monitoring events. Examples:

- A recipe described as "Neighborhood Watch" that sends a notification whenever an Instagram photo has been taken in a given area.
- A recipe described as "Flash my lights when my upload to Vimeo is complete"
- A recipe that sends a text message to the user when there is breaking news about a football team.

Level 4- User does not see the outcome of an action until he or she explicitly seeks it out.

Many level 4 recipes were similar to level 1 in that they are mostly used to *archive* information or data artifacts such as photos. However, in level 4 recipes, the user either has no involvement in triggering the recipe, or the trigger is a surveillance of the user's natural behavior outside of interaction with their technology (i.e. something the user must do whether or not they are using a system, and is not necessarily done specifically to trigger the recipe).

- A recipe that saves new photos on Instagram by a specific (other) person into a new note on Evernote.
- A recipe, described as "Log hours at work" that records the time on a spreadsheet whenever a phone enters or exits an area.
- A recipe described as "Log snow in your city" that records data from a weather RSS feed to a spreadsheet.

Level 5- User is completely out of the loop.

In these recipes, the user has no explicit or intentional involvement in triggering the recipe, in the action that is triggered and there is no clear followup action. These recipes are designed to make apps and devices work invisibly around the user, typically eliminating the need for interaction or reducing the burden of a repetitive process. Examples:

- A recipe described as "Mute at school" that turns off a phone's volume when connecting to a specific WiFi network.
- A recipe described as "Turn off heating on warm days" that shuts off a furnace when a weather feed indicates a particular temperature.
- A recipe that automatically posts breaking news about a sports team to Facebook.

I found several distinct purposes for recipes at this highest level of automation. Many recipes served to reduce or eliminating a burdensome repetitive task, such as backing up files in one cloud storage service into a different cloud storage service, or turning on lights at sunset.

A theme found in many level 5 recipes was preventing or correcting human error. For example, one recipe connected the users calendar and home lighting system in order to "make sure my lights are off when in a meeting". Another recipe checked to be sure that the garage door was opened when a car's ignition was turned on. And recipes were sometimes used to bypass or correct the behavior of people other than the user, such as a recipe connecting a trigger and action both from a thermostat in order to "stop wife turning heating up to max." Similarly, dozens of recipes appeared to be useful for maximizing phone battery life and ensuring that the ringer volume is appropriate for a particular context/location.

Another theme was automating communications. Many recipes were set to automatically wish people happy birthday or Merry Christmas, or text "Good Morning to the Girlfriend."

Table 2. Level of automation of IFTTT recipes.

Level	1	2	3	4	5
Number	30	36	57	45	82
%	12%	14%	23%	18%	33%
Inter-rater reliability (Cohen's Kappa)					0.538

Summary

Table 2 describes the number of recipes in each level of automation. Level 5 recipes were most common, although the majority of these were recipes that automatically share information to social media. Overall however, these fully automated recipes only account for a third of all automations in the dataset. While high degrees of automation are useful and common, users nonetheless largely keep themselves in the loop of their automations in some capacity, using them primarily to manage information than to execute actions.

USER STUDY OF HEALTH AUTOMATIONS

To better understand how users of health information web services might make use of personal automations, I designed a platform to mimic the interface of IFTTT.com, but included action and trigger channels that are primarily related to health. This platform was not functional, in that it could not actually connect to external web services or create any working automations. Its purpose was to provide an end-user programming interface that would enable users to create hypothetical automations, and then collect data from the user about each automation's intended purpose and benefit in their personal health information management.

In consultation with healthcare workers, physicians, patients, and health services researchers, I brainstormed a set of channels that might contain health information or have relevance to health. These included things like an online patient portal, accounts with a pharmacy or insurance company, a personal health record like Microsoft Health Vault, connected medical devices and sensors such as fitness trackers, at-home blood glucose monitors or fertility monitors, and other major online sources of health information such as WebMD, Drugs.com, and ClinicalTrials.gov. In addition to these health-specific channels, I included several important channels that I noted in the analysis of IFTTT, such as email and messaging, phone API's, social media sites, calendars and cloud file storage services.

In creating triggers and actions for these channels, I referred to the initial analysis of triggers and actions from IFTTT and sought to create numerous triggers and actions that fit in each of the categories, such as *data* triggers or *activate* actions.

Table 3. **Health Automation Trigger-Action Connections. Values indicate number of recipes for the combination of trigger and action types. Note that triggers and actions could have more than one type.**

	Action					
Trigger	Notify	Record	Share	Activate	Adjust	Total
Value	31	22	21	8	1	73
Change	10	5	8	5	0	28
Time	26	15	15	7	0	63
Data	54	30	39	12	2	127
News	3	1	3	1	0	8
Sensor	16	10	12	6	1	45
Location	1	0	1	0	0	2
Total	141	83	99	39	4	

Table 4. Statistics and examples of health automations by level of automation

Level	Count (%)	Example
1	6 (3%)	Sending a text message to post to Facebook
2	93 (46%)	Send user an email when a new test result is available in patient portal
3	25 (12%)	Creates a Facebook post when a daily goal for steps has been reached
4	18 (9%)	Record any new blood glucose measurement to Microsoft Health Vault
5	59 (29%)	Automatically pay a new bill
Inter-rater reliability- 0.68 (Cohen's kappa)		

The platform overall had 31 distinct channels with 102 different triggers and 43 different actions available.

I recruited 34 participants for the user study from a clinical studies participant registry at a university health system, and compensated participants with a $5 gift card. I screened interested participants using a questionnaire to assess their familiarity and competence using web applications and experience using advanced software features such as setting up an email filter or writing a computer program. I selected participants who indicated a high degree of "tech savviness", as I wanted participants who would be familiar with the channels available on the site and would feel confident using an end-user programming interface to create hypothetical automations. 55% of participants were female. Ages ranged from 18 to 69, with a median age of 29.

Figure 3 shows the interface for creating an automation on the platform. Participants were instructed to create any automation that they would find useful in their daily lives or for managing their health. Many triggers and actions supplied fields to add qualifiers to the trigger (e.g. indicate who an email should be sent to). Since these automations were hypothetical, participants were instructed to provide qualifiers that would be meaningful to the researchers in analyzing the recipe (e.g. "send an email to [my mother]"), rather than a specific value that would only be useful if the automation were live (e.g. "send an email to [email@example.com]").

After creating an automation, participants were asked to provide a statement about the purpose and benefit of the automation. Participants were asked to create at least 5 automations, although they were permitted to create as many as they liked. The final dataset includes a total of 201 automations.

Analysis

Table 3 shows how many recipes were created connecting each type of trigger and action. *Data* triggers, *notify* actions were the most common in the data, as were automations connecting those types. As with the IFTTT data, the higher-automated *activate* and *adjust* actions were less used than the other types.

I again coded each automation using the level of automation schema described above (see Table 4 for statistics from the coding).

Level 5 automations that keep users out of the loop accounted for only 29% of all automations. While this is more than what was observed on IFTTT, it still suggests that users largely want to keep themselves in the loop in personal health automations. Level 2 automations that primarily sent reminders were most common and accounted for nearly half of the health automations. As monitoring systems or information is a task that is far better performed by computers than by humans [25], this represents an efficient use of automation to aid in personal health management and decision making.

I again analyzed each recipe looking both to find themes consistent with what was found in IFTTT and also seeking out new themes and concepts that may be specific to health contexts of personal automations.

As with the IFTTT dataset, I found examples of using automation to reduce repetition, alerting the user of something they need to do, informing them of changes in the world, communicating with others, and archiving information. Table 5 describes some examples from each category of automation.

Some automations served to create behavioral interventions for the user. For example, one participant created an automation that automatically orders healthy foods according to a calendar, stating:

> "When I get paid, I sometimes decide to go and eat out, at which times I do not make the healthiest choices. If there would be a way to automatically place an order for healthy foods on payday, then it takes the decision to purchase unhealthy foods out of the equation."

Another automation sends a text message encouraging the user to buy produce when entering a grocery store. And many automations were used to create motivations to walk more steps or eat healthy foods by posting messages indicating positive or negative behaviors to social media.

Also in similarity to the IFTTT recipes, many automations could be described as a means to simply move information to a different interface. 58% of all automations in the user study either sent the user an email, a text message, or placed something in the user's calendar. These automations moved information from the various corners of the web-based health

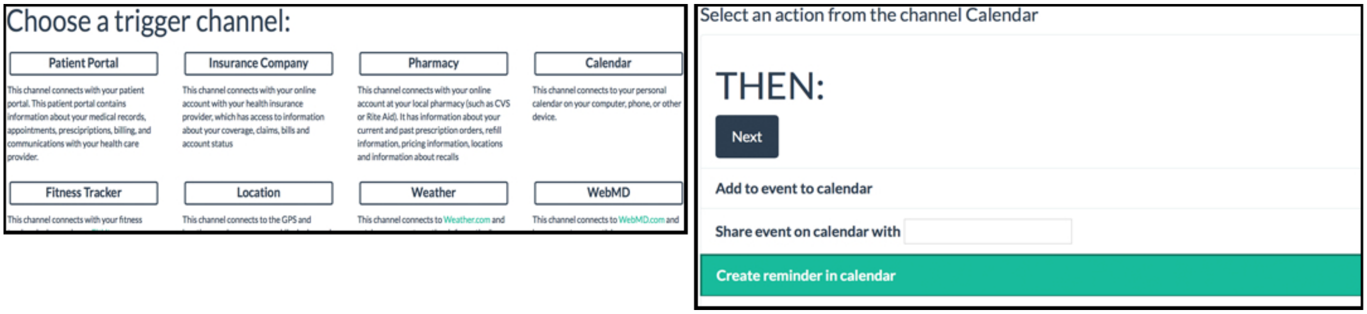

Figure 3. Interface for the prototype integration platform for health.

Table 5. Personal Health Automations examples

Automation purpose	Examples
Consolidate Interfaces	When new test result is available, send text message containing the result
	”To avoid logging in to 23andMe,” create text file of new report in cloud storage.
Archive Information	When insurance bill is posted, add new row to spreadsheet containing the bill amount
	Add new blood glucose measurement to Microsoft HealthVault
Regulate behavior	Post message to Facebook if I oversleep to ”motivate me”
	Place order for fruit if new weight is greater than 2 pounds my previous weight
Share information	If a blood glucose measurement is too high, send a message to my doctor
	If a new clinical trial related to asthma is posted to ClinicalTrials.gov, post a link on the asthma forum at PatientsLikeMe.com
Get informed	Email me any new article on WebMD.com about autoimmune disorders
	Email me when someone else accesses my health record
Alert or remind oneself	Send me a text message if a current prescription needs to be refilled
	Email me if an insurance bill is due
Reduce burden	Turn on air conditioning if humidity rises above 80%

data ecosystem into a very limited number of information interfaces.

I found evidence that users can create personal automations to help them navigate through the cracks in a complex healthcare system. For example, a participant created an automation to immediately file a claim any time a prescription is ordered, stating

> “Because we have prescription medication coverage under two different health insurance plans, we now have to file a prescription claim ourselves with our secondary insurance when the primary does not cover a prescription or only partially covers it. This would be helpful with both maintenance prescriptions that are ordered thru an online service or with one time prescriptions ordered thru a local pharmacy.”

Many other recipes were designed to automatically set up an appointment or send a message to a doctor if various sensors recorded values above a certain for things like blood glucose, blood pressure, or body temperature. In these cases, patients might use the end-user programming of an integration platform to codify decision heuristics about when to talk to a physician.

DISCUSSION

These analyses show that people can use integration platforms to create automations that perform many important personal informatics tasks. On both a general integration platform and a health-focused platform, users created personal automations to:

- Consolidate different services into a single interface
- Archive information
- Enforce or regulate desired behaviors of the user or others
- Prevent or correct errors
- Share information with others
- Inform oneself about new information
- Alert or remind oneself of a needed action
- Reduce or eliminate a burdensome or repetitive task

These findings illustrate that there are many tasks within the requirements of effective personal health informatics that are good candidates for automation. Monitoring patient portals for new test results, reminding users to take important actions such as refilling prescriptions or paying a bill, or keeping up to date with new clinical trials are functions that small automations can provide clear value to people in managing their health information and supporting health behaviors and decisions. Naturally, many of the systems that became integrated with each other through the platforms have built-in functions and features for configuring many of these types of automations. Alerts or reminders are particularly common feature in all kinds of apps and devices, as there is significant need for systems to monitor events and information and only involve the user when necessary. It is therefore not surprising that a large proportion of the automations in both datasets were alerts or reminders.

However, even the unsurprising findings about alerts illustrate how the integration platform model for health informatics adds flexibility to the design of health information technologies. While many automations served to alert the user of something, there was great heterogeneity in terms of the chosen mode of delivery. Some automations send emails, others send text messages, others create calendar alerts, or others create various in-app notifications. The integration platform model offers customizability without significant need on the part of system designers to foresee all possible avenues for alerts and designing them into the system in a top-down way. Rather, the integration platform model affords a more bottom-up approach to personal informatics design that is flexible and participatory.

Using automations to share information with others also highlights the value of an integration platform model in health. Chinta and Raghaven [7] have argued that PHR's are not "social enough" and that many users have desires to treat a PHR much like a social networking site, and other work has found that the ability to share and manage access to health information with others is a necessity for effective health information tools [9][27]. The integration platform model both offers a capacity to extend sharing capabilities to systems that may not otherwise have it built in (such as insurance company accounts), or offer flexibility in the way information is shared.

An unexpected but important finding was the use of personal automation to create behavioral interventions such as automatically ordering healthy foods on payday to reduce temptation or automatically posting to social media when an exercise goal is not reached so as to create accountability. These findings illustrate that the integration platform model enables personal informatics to go beyond mere management of health data and become active tool in regulating healthy decisions and behaviors.

Klasnja et al. [19] describe the the issue of "unanchored information activities" that are prevalent in health information management. Interaction with health information can happen at unpredictable times or locations, and access to tools may be limited at the moments they are most needed. For example, users may need to see their calendar's when scheduling doctor's appointments but they may not be available at the needed moment. Klasnja et al. find that even users of mobile technologies for health management desire to consolidate their interaction with health information. For example, in their study, users expressed a strong desire to have health-related appointment calendars simply integrate with their regular calendars. The findings from my study corroborate those findings, as many automations were used to simply push information from an app or website into a calendar, email, or text messaging app.

Automating the consolidation of health information into a small number of interfaces, primarily those available through mobile devices, can help to mitigate the problem of unanchored information tasks.

These studies show that when given a powerful integration platform, users are capable of combining different systems and creating automations to fill important needs for personal information management and for supporting health and wellness. This model for personal informatics design can be empowering and participatory by giving users control over how and when they see (or don't see) information, as well as the opportunity to concede direct control to systems in order to minimized burdens or enforce behaviors.

Implications for Design

An important implication for the design of health technologies is that there is value in designing for integration. While stand-alone user interfaces for health apps or devices will continue to be important, designers should consider how users might use automated integrations to interact with a system's information through a more general personal information management tool like email, text messaging, calendars, or operating systems. Integration platforms make it relatively easy to automatically move information from one interface into a more preferred interface. As noted by one participant:

> "Everything I do runs from my phone calendar and, while I can input things, automation of that makes everything simpler."

Therefore, designers must consider how a systems information may be used outside the context of the system itself. Designers must consider satisfying important usability heuristics such as making system status visible or helping users recover from errors [24] when users are interacting with their systems indirectly via other apps or platforms.

Other important implications derive from notable omissions within the personal automations in these datasets. An important area of research in HCI for health and wellness has focused on personal data tracking or the "quantified self" [8]. However, users often have trouble making sense of the data collected via sensors or other automated means [18]. The automations used in these studies often served to collect data and make it accessible or available when needed. But there were no automations that provided any kind of support for understanding what automatically-collected data means for one's health. Integration platforms have an opportunity to design ways not only for users to get more data, but also to make better sense of it and use it to inform decisions.

Also notably missing from these datasets were automations that execute an action automatically, but provide users an opportunity to stop or alter the automation before executing the action. For example, in automations that automatically pay an insurance bill, many users might want to be able to have the bill sent to them via text message, and be able to send a reply to stop the automatic payment if something is unsatisfactory. Sheridan and Verplank's levels of automation taxonomy [28] includes similar levels of automation, but the taxonomy I used in these studies did not because I found no examples of this type of automation. This is a consequence of the simple affordances of the IFTTT platform (and the health-focused clone) that do not permit complicated automations using multiple triggers [14]. While in theory this might be accomplished via combinations or chains of automations, it is not clear whether users do this, as an analysis of a single user's automation collection was not possible in these data. It is clear that there is a design tradeoff between the simplicity and usability of the end-user programming interface and the capabilities afforded by the platform. However, adding capabilities for multiple triggers or other ways to keep users even more lightly in the loop of personal automations will enable greater richness in personal automations and enable even greater satisfaction of user needs.

There are additional implications for sociotechnical theories of health decision making. The integration platform model makes users participants in their personal health informatics design. Much like a personal health "untoolkit" described by Jelen and Siek [16] that enables users to craft their own devices, integration platforms allow users to design for themselves.

However, designing interactive systems can be a challenging task even for those with great experience. While users may have exceptional understanding of their own needs and preferences, they may lack the capacity to formulate an effective design to meet those needs. Programming an automated routine requires users, at the time of creating or adopting an automation, to anticipate the circumstances and preferences that will exist when they automation executes. End-user programmers often do not do this well [14].

For example, many automations in the data sent text messages when some type of data may be available, such as a new test result in an online patient portal. If the data become available in the middle of the night, would users really want to be receiving those text messages at that time? Furthermore, research on how people form preferences suggests that people construct preferences on the fly as needed, rather than stored and retrieved [31]. Will the users construct the same preferences when programming an automation as they would when actually using the automation? Many behavioral intervention automations made use of this disconnect, such as those that automatically order healthy food or vitamins. These automations seem to leverage an assumption that rule-based decisions made a priori through programming an automation will be more rational or are better aligned with one's true preferences than those made in the heat of a moment. But not all users are likely to think in this manner, and in many cases mismatches between preferences at the time of programming an automation and the time of using it may create frustrations for many users.

The findings suggest that creators of personal automations will not necessarily make well-informed user-centered design decisions simply because they themselves are the user. Consider the large number of automations for alerting the user of some event, and the fact that many participants in the health study created *only* alerting automations. These users likely have a desire to stay tightly in the loop of their information management, but an overuse of alerts may lead to problems of alert fatigue or information overload.

Or consider the commonly created automations that automatically refill prescriptions. While such an automation reduces the burden of remembering to fulfill a prescription, it subsequently creates a new burden of remembering to stop the automation when that becomes necessary (aka the "irony of automation" [2]). While sophisticated automations could be crafted to further minimize these types of potential usability issues, integration platforms need to assist users in recognizing and carefully planning their own role in the human-machine partnership. A critical area for future research will be to design integration and end-user programming interfaces that teach users to not just be designers but to be good user-centered designers.

Limitations and Future Directions

There are many limitations with these studies that demand attention. First, participants may have created automations in the user study that seem useful in the context of a study but that they would never actually use in real life. The numerous technical and economic challenges to creating a functioning integration platform, or even a functioning integrated personal health record [7], mean that there is no data available about how people would actually create working personal automations for health. Although some insight has been gained from analyzing the real automations in IFTTT, there is a clear need to find ways to study how people create and use working personal health automations.

The participants in the user study, although recruited from a university health care system, were screened for a relatively high degree of "tech savviness" and experience using networked computing. The population of IFTTT users likely skews towards highly tech-literate users. In order to make the integration platform model effective and equitable, future research must consider how to make the personal automations accessible and understandable for users at all levels of digital literacy.

CONCLUSION

This exploration of personal automations illustrates that Integration Platforms show considerable promise as a model for empowering people to connect fragmented health information technologies to expand functionality and fulfill needs for personal health information management. Much work is required, however, to design ways to help people optimize the potential of these automations.

ACKNOWLEDGEMENTS

Withheld for review

REFERENCES

1. Neeraj K Arora, Bradford W Hesse, Barbara K Rimer, Kasisomayajula Viswanath, Marla L Clayman, and Robert T Croyle. 2008. Frustrated and confused: The american pub-

lic rates its cancer-related information-seeking experiences. *Journal of general internal medicine* 23, 3: 223–228.

2. Lisanne Bainbridge. 1983. Ironies of automation. *Automatica* 19, 6: 775–779.

3. AJ Brush, Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon. 2011. Home automation in the wild: Challenges and opportunities. *Proceedings of the sigchi conference on human factors in computing systems*, ACM, 2115–2124.

4. Federico Cabitza, Daniela Fogli, Rosa Lanzilotti, and Antonio Piccinno. 2015. End-user development in ambient intelligence: A user study. *Proceedings of the 11th biannual conference on italian sigchi chapter*, ACM, 146–153.

5. Jill Cao, Yann Riche, Susan Wiedenbeck, Margaret Burnett, and Valentina Grigoreanu. 2010. End-user mashup programming: Through the design lens. *Proceedings of the sigchi conference on human factors in computing systems*, ACM, 1009–1018. http://doi.org/10.1145/1753326.1753477

6. Nicholas Carr. 2014. *The glass cage: Automation and us*. W.W. Norton; Company, New York.

7. Ravi Chinta and Vijay V Raghavan. 2015. Phenomenological study of decline of personal health records: Empirical evidence from thematic analyses of blogs' content. *Cogent Business & Management* 2, 1: 1102617.

8. Eun Kyoung Choe, Nicole B Lee, Bongshin Lee, Wanda Pratt, and Julie A Kientz. 2014. Understanding quantified-selfers' practices in collecting and exploring personal data. *Proceedings of the 32nd annual acm conference on human factors in computing systems*, ACM, 1143–1152.

9. Andrea Civan, Meredith M Skeels, Anna Stolyar, and Wanda Pratt. 2006. Personal health information management: Consumers' perspectives. *AMIA*.

10. Luigi De Russis and Fulvio Corno. 2015. Homerules: A tangible end-user programming interface for smart homes. *Proceedings of the 33rd annual acm conference extended abstracts on human factors in computing systems*, ACM, 2109–2114.

11. Don Detmer, Meryl Bloomrosen, Brian Raymond, and Paul Tang. 2008. Integrated personal health records: Transformative tools for consumer-centric care. *BMC medical informatics and decision making* 8, 1: 1.

12. Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. 2003. The role of trust in automation reliance. *International Journal of Human-Computer Studies* 58, 6: 697–718.

13. Karen M Feigh and Amy R Pritchett. 2014. Requirements for effective function allocation a critical review. *Journal of Cognitive Engineering and Decision Making* 8, 1: 23–32.

14. Justin Huang and Maya Cakmak. 2015. Supporting mental model accuracy in trigger-action programming. *Proceedings of the 2015 acm international joint conference on pervasive and ubiquitous computing*, ACM, 215–225.

15. Alejandro R Jadad, R Brian Haynes, Dereck Hunt, and George P Browman. 2000. The internet and evidence-based decision-making: A needed synergy for efficient knowledge management in health care. *Canadian Medical Association Journal* 162, 3: 362–365.

16. Ben Jelen and Katie Siek. 2016. Creating a health sensing crafting untoolkit and methodology. Retrieved from https://drive.google.com/open?id=0By7DbUkuvUE8bWZobXJ5bnNmVWs

17. Jakob D Jensen, Miao Liu, Nick Carcioppolo, Kevin K John, Melinda Krakow, and Ye Sun. 2016. Health information seeking and scanning among us adults aged 50–75 years: Testing a key postulate of the information overload model. *Health informatics journal*: 1460458215627290.

18. Simon Jones and Ryan Kelly. 2016. Sensemaking challenges in personal informatics and self-monitoring systems.

19. Predrag Klasnja, Andrea Hartzler, Christopher Powell, Giovandy Phan, and Wanda Pratt. 2010. Health weaver mobile: Designing a mobile tool for managing personal health information during cancer care. *AMIA annual symposium proceedings*, American Medical Informatics Association, 392.

20. John Lee and Neville Moray. 1992. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics* 35, 10: 1243–1270.

21. Henry Lieberman, Fabio Patern, Markus Klann, and Volker Wulf. 2006. End-user development: An emerging paradigm. In *End user development*. Springer, 1–8.

22. Robert Molloy and Raja Parasuraman. 1996. Monitoring an automated system for a single failure: Vigilance and task complexity effects. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 38, 2: 311–322.

23. Michael J Muller and Sandra Kogan. 2010. Grounded theory method in hci and cscw. *Cambridge: IBM Center for Social Software*: 1–46.

24. Jakob Nielsen. 1995. 10 usability heuristics for user interface design. *Fremont: Nielsen Norman Group.[Consult. 20 maio 2014]. Disponvel na Internet*.

25. Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 39, 2: 230–253.

26. Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. 2000. A model for types and levels of human interaction with automation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 30, 3: 286–297.

27. Wanda Pratt, Kenton Unruh, Andrea Civan, and Meredith M. Skeels. 2006. Personal health information management.

Commun. ACM 49, 1: 51–55. http://doi.org/10.1145/1107458.1107490

28. Thomas B Sheridan and William L Verplank. 1978. *Human and computer control of undersea teleoperators*. DTIC Document.

29. Thomas B Sheridan. 2000. Function allocation: Algorithm, alchemy or apostasy? *International Journal of Human-Computer Studies* 52, 2: 203–216.

30. Thomas B Sheridan. 2006. Supervisory control. *Handbook of Human Factors and Ergonomics, Third Edition*: 1025–1052.

31. Dan Simon, Daniel C Krawczyk, and Keith J Holyoak. 2004. Construction of preferences by constraint satisfaction. *Psychological Science* 15, 5: 331–336.

32. Dean F. Sittig. 2002. Personal health records on the internet: A snapshot of the pioneers at the end of the 20th century. *International Journal of Medical Informatics* 65, 1: 1–6. http://doi.org/http://dx.doi.org/10.1016/S1386-5056(01)00215-5

33. Linda J. Skitka, Kathleen L. Mosier, and Mark Burdick. 1999. Does automation bias decision-making? *International Journal of Human-Computer Studies* 51, 5: 991–1006. http://doi.org/10.1006/ijhc.1999.0252

34. Paul C. Tang, Joan S. Ash, David W. Bates, J. Marc Overhage, and Daniel Z. Sands. 2006. Personal health records: Definitions, benefits, and strategies for overcoming barriers to adoption. *Journal of the American Medical Informatics Association* 13, 2: 121–126. http://doi.org/10.1197/jamia.M2025

35. Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L Littman. 2014. Practical trigger-action programming in the smart home. *Proceedings of the sigchi conference on human factors in computing systems*, ACM, 803–812.

36. Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, et al. 2016. Trigger-action programming in the wild: An analysis of 200,000 ifttt recipes. *Proceedings of the 2016 chi conference on human factors in computing systems*, ACM, 3227–3231. http://doi.org/10.1145/2858036.2858556

37. Jeffrey Wong and Jason I Hong. 2007. Making mashups with marmite: Towards end-user programming for the web. *Proceedings of the sigchi conference on human factors in computing systems*, ACM, 1435–1444.

38. Allison Woodruff, Sally Augustin, and Brooke Foucault. 2007. Sabbath day home automation: "It's like mixing technology and religion". *Proceedings of the sigchi conference on human factors in computing systems*, ACM, 527–536. http://doi.org/10.1145/1240624.1240710

39. Pamela Zave, Eric Cheung, and Svetlana Yarosh. 2015. Toward user-centric feature composition for the internet of things. *arXiv preprint arXiv:1510.06714*.