EPIC2016

Papers 5 – Analytics

Paco—Applying Computational Methods to Scale Qualitative Methods

BOB EVANS
Google, Inc.

For several years we have been building and using an open mobile research platform, called Paco, that enables the scaling of qualitative research through quantitative, computational techniques. The platform provides a mechanism to design and deliver remote research instruments to mobile devices in the field and it provides mechanisms to abstract and develop new research tools.

The most immediate way the platform scales qualitative research is by enabling researchers to visually design, deploy, and manage research instruments comprised of surveys, triggers and sensor logging without needing to program or build a new mobile app. The combination of sensors, surveys and triggers supports idiographic, phenomenological, qualitative inquiry as well as contextual data collection in participants' natural setting.

Stepping back, the platform is an experiment in scaling the generation of new instruments and the generation of new knowledge in the science itself. Under the covers, Paco is implemented as an open construction kit of research components for all to use and modify as they like. Its design enables computational thinking [Wing] about both qualitative and quantitative behavioral research. It makes it possible to generate an infinite combination of research instruments from basic building blocks. Specifically, it borrows concepts and practices from programming language design, software architecture, and the software community.

By writing down the elements and methods for research instruments in a precisely specified, machine-executable language, they become more clear. This makes them better understood. This scales the generation of knowledge in behavioral science.

Automated tools cannot replace the researcher. Ethnography has a very deep rich practice of immersive field work and analysis. The researcher is the ultimate instrument for understanding what is significant both individually and culturally within any study. Paco merely offers tools to support and advance the practice by scaling methods, automating parts that are amenable, and, by facilitating precise characterization of the data and data collection protocols.

There are many challenges to how well computational methods can model and support behavioral research, particularly the qualitative methods used in ethnography. We finish with a discussion of some of the theoretical and practical challenges and how our method meets, and doesn't meet, those challenges.

INTRODUCTION: SCALING QUALITATIVE INPUT WITH COMPUTERS AND COMPUTATIONAL PRACTICES

For several years we have been building and using an open mobile research platform, called Paco, to conduct qualitative and quantitative research into user experiences in daily life, including experiences of technology such as mobile phones as well as health and wellness. The platform provides a mechanism to design and deliver remote research instruments to mobile devices in the field and has been used by over 400 researchers to create over 4300 experiments involving over 25000 participants.

2016 Ethnographic Praxis in Industry Conference Proceedings, p. 348–368, ISSN 1559-8918, https://www.epicpeople.org

Most importantly, the platform uses computational methods to allow reasoning over the research methods themselves. This allows development of new higher-level methods from existing components and recombination of components to generate an infinite number of new, more precise research tools from basic building blocks. It does this using an open scientific method that allows sharing and improvement of knowledge in the research community.

Ethnographic research, and the qualitative methods it employs in particular, are capable of producing deep, rich insights about individuals and their culture. Leading corporations have sought out this valuable insight into their customers. The EPIC literature is full of examples. One example in particular is the IBM CEO study of 2010 which reveals that CEOs see the business environment becoming more complex and that they fear they will not be able to operate effectively in that environment. They look to ethnographic methods to help them better create meaningful products and services for their customers. As Roger Martin, dean of Rotman business school puts it, “"Ethnography is essential to Innovation” [Ladner].

So, why aren’t all businesses clamoring to ethnography for insights into their users? One challenge is that traditional ethnography with immersive field work and a long analytical process after the fact is costly and takes a lot of time. Sam Ladner also lists another possible cause. The essentially “operational, quantitative” orientation of business is in opposition to the “descriptive, qualitative” nature of ethnography. Business is quantitatively oriented because that is the primary approach they have used to increase their success [Ladner].

This dilemma between expensive, rich descriptive data and quantitative operations might be more usefully framed as an example of the classic explore-vs-exploit dilemma [Christian]. Is it worth it to expend energy looking in new uncertain directions for opportunities or is it better to keep the nose to the grindstone and exploit the present, clearly understood opportunity? CEOs from the study above understand that they need to explore but it goes against the quantitative culture which optimizes exploiting the current opportunity.

Clayton Christensen proposes an answer in his classic business book, *The Innovator’s Dilemma* [Christensen]. Successful businesses have to hedge against the inevitable commoditization of their current “cash cow” product line by investing some amount of their resources in finding the next successful product. He recommends allocating some significant fraction of resources to exploring new options and the majority to maintaining the current offerings.

This is a useful way to think not only about whether to invest in research such as ethnography and qualitative methods but also about how to incorporate these methods. Is there an approximation of the ethnographic research value proposition that can provide more of the “emic” stance, thus making products that are more meaningful to users than they otherwise would have been without the full cost of an ethnographic research project? This paper takes the position that it is possible to approximate some of the methods of ethnography through technological aids and that this will provide useful insights unavailable otherwise. It also takes the position that we can improve these approximations using computational thinking and methods.

Gathering qualitative, first-person experience using a theoretically grounded method along with collection of contextual information from sensors all within the person’s natural context can provide individual as well as cultural insights. With some care, it is possible to craft research instruments in a way that respects ethnographic principles thus enabling the synthesis of cultural insights.

The researcher still needs to ensure their method is ecologically valid and that their approach is ethnographically rooted, i.e., focused on investigating the culture through the norms and outliers presented in behaviors, feelings, preferences, and affinities of individuals. Additionally, gathering as much data about the context and use of artifacts by individuals is important in developing a full description of the culture.

Identifying Parts of Qualitative Methods to Approximate

There are several qualitative methods and concepts from ethnography that the Paco research platform supports. There is the interview, of course, in the form of survey questions which also allow branching with simple or complex predicate logic to allow the interview to proceed along paths known to the researcher to be meaningful.

The survey can also be used as a structured interview to get within-person repeated measures or across-person measures for understanding the range of responses in a culture. Repeated measures through Experiential Sampling (ESM) [Hektner] can help quantify qualitative responses and may help achieve "saturation" [Glaser], the ethnographic concept where the distribution of the occurrences of a phenomenon is complete enough to point to norms and outliers.

With passive or active collection of data from phone sensors of physical actions and states in the environment it becomes possible to get a broader picture of where they are in their environment and how they uses artifacts such as transportation and mobile phones in their daily lives. This is an approximation of some aspects of participant observation as well.

By being delivered via mobile phone, the data is collected in the person's natural setting at the salient moments, thus providing opportunities to get better ecological validity. By using the sampling methods provided, experiential sampling schedules and event-triggered sampling, it becomes possible to further improve the ecological validity and reduce other biases in user responses.

There is another benefit of collecting the data without the researcher being present. While normally, the researchers presence in the environment allows a very rich data set compared to a remote sensing application on a phone, the researcher's presence may introduce changes in behavior and in what is observable due to the researcher's social role [LeComte]. The phone app may be less intrusive for some types of data collection and it is hypothesized that its social role will be less dynamic.

The Paco platform provides primitives for signaling the user and asking questions as configurable components so that they can be arranged as desired to target particular moments of interest and to get data when it most salient and least erroneous. The experiential sampling method is a particular combination of these components, using random sampling and a focus on momentary experiences. It is based in the philosophical school known as Phenomenology [Phenomenology].

This is a theoretical framework that could be very much said to take the "emic" stance of ethnography. It puts the individual's thoughts, feelings and experiences first. Phenomenology says that the person's experiences, mediated by the body and the senses, and the things to which they pay attention determine their stance toward the world. This stance determines their beliefs. Their beliefs in turn determine their actions. In some sense, perception becomes reality. In practical private sector ethnography, this is how we can better understand what users find meaningful and why they do or do not adopt products and services.

Circumstances That Made Paco Possible

The Paco platform started as an engineering tool to understand the dimensions of productivity in an commercial software engineering setting. Given the high cost to industry of recruiting and retaining engineers, it was important to understand what factors contributed to productive, engaging work for engineers under the hypothesis that an engineer in such a state would stay on and contribute productively to the organization.

The key technological innovation that made Paco possible was the introduction of the smartphones such as the iPhone and the Android operating system. The smart phone combined the capabilities of a full general-purpose programmable computer with an array of sensors and actuators in a small enough package that it is carried by participants throughout their day.

By being able to sense and record aspects of the context in which the participant lives, it is possible to build a more comprehensive picture of the natural setting than could be done with a traditional survey instrument. It is certainly not the same as an ethnographer in the natural setting but it offers a different set of data.

By being able to actuate in the participant's environment, e.g., signaling the participant to respond to questions, it can facilitate collection of data at moments of interest when they are most salient.

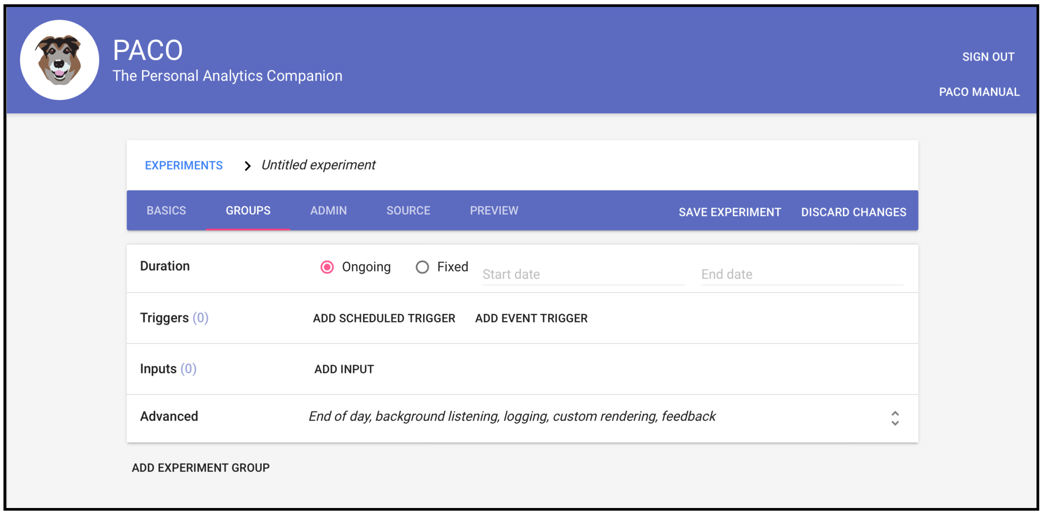
The power of the general purpose computer is the ability to build an infinite number of software "machines". In Paco's case, this means the ability to compose sensing and actuating components into an infinite number of behavioral research instruments. It is this compositional power that Paco tries to maximize in order to scale research.

PACO'S RESEARCH CAPABILITIES THROUGH THE VISUAL EXPERIMENT DESIGNER

Paco [Paco] provides visual facilities (see figure 1) for designing, deploying, and monitoring experiments, and exporting data to statistical analysis tools. It does this with strong security and clear privacy controls.

The visual editor enables researchers who have no ability to program to create a wide variety of research instruments. If more power is needed, Paco provides a custom programming library in Javascript that most CS students could easily master. The strategy is to incorporate the most commonly used elements of research protocols available into the visual editor so the largest number of researchers can use them without resorting to custom programming.

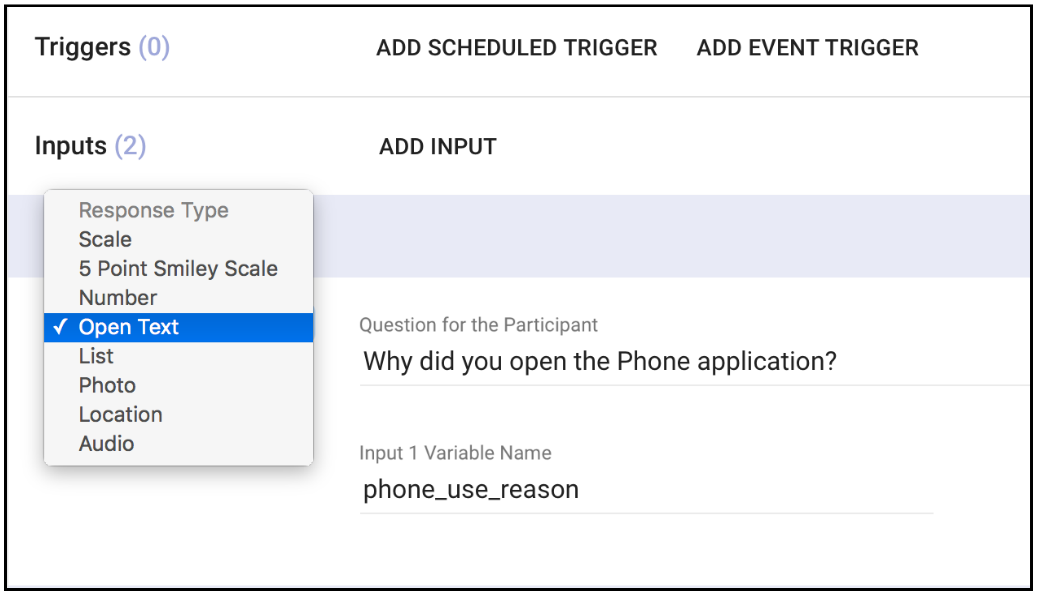
Paco's first qualitative method was an implementation of the ESM on Android and iOS smartphones but it has added many other functions for capturing data. In this first section, we will expound on the features for designing and deploying research instruments that enable scaling the execution of research.



Duration ● Ongoing ○ Fixed Start date End date

Advanced *End of day, background listening, logging, custom rendering, feedback*

The two main activities when designing a research instrument are predicated on the research questions to be investigated. The first activity is to determine what data should be collected to inform the research question and the second activity is to determine when to collect it so that the sample is as strong and as valid as possible.



✓ Open Text

Question for the Participant
Why did you open the Phone application?

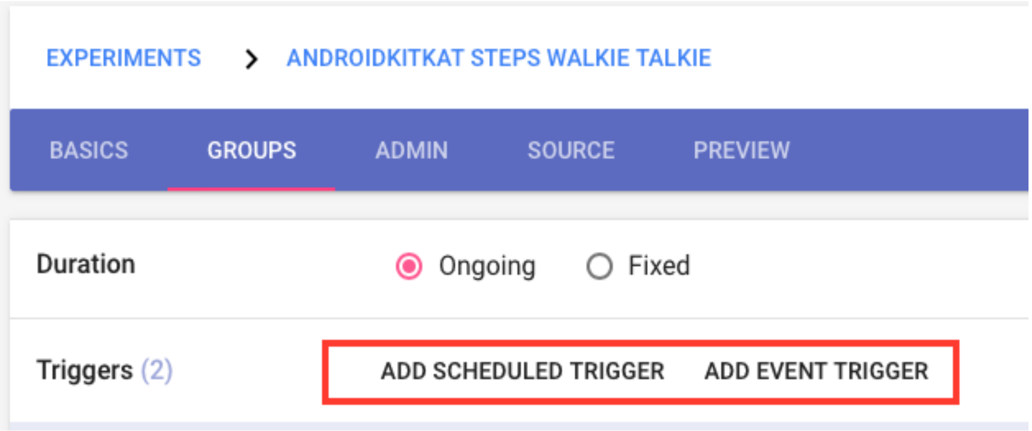
Input 1 Variable Name
phone_use_reason

Types of Data Collection Supported

Paco allows a researcher to design surveys and to specify sensors from which to collect data. Sensors include location, app usage, fitness metrics (step counts), calendaring and other phone behaviors. For surveys, Paco supports qualitative and quantitative items. It offers open text items, photo items, and audio recording items. It also offers scale items, list items (single and multiple selection), numerical items, and location (latitude and longitude).

In addition, surveys have a powerful branching logic system that allows questions to be asked only when specified criteria are satisfied. In contrast to normal skip logic which only specifies which part of the survey to jump to, Paco, specifies what criteria should be true before asking a question. This allows the researcher to easily specify multiple criteria for a given question or line of questioning.

Also, with Paco it is possible to have several different instruments in a study. For instance, there may be one set of questions asked as a pre-questionnaire, another as an ongoing daily study, and another that passively records sensor data at specified intervals.



Duration ● Ongoing ○ Fixed

Triggers (2)

Defining the Moments to Collect Data

Paco, as mentioned supports randomized sampling, such as ESM. It also supports fixed interval signaling, e.g., Daily, Weekly, Monthly. Additionally, sampling can be event-contingent. This is defined as states or actions that can be sensed on the phone. Examples of events that can be sensed and thus used as a trigger include a user opening their phone, taking a phone call, using a particular application, entering a certain geolocation, or playing music. When one of these actions occurs, Paco can trigger them to participate in a survey.

Event triggers and scheduled triggers can be also be combined. One of the big challenges with remote data collection is getting good data. There are several factors that can affect data quality. One is getting the sampling to match the underlying effect frequency. Having more ways to cover time and events makes it more likely that an instrument can be built to collect data when those moments are still salient. If the moment has passed, a user may no longer recall it, or may recall it incorrectly.

Paco also allows setting a limit on the time allowed to respond. If the user is given a lot of time to respond or knows when they are going to be asked they may prepare more socially acceptable answers. For a detailed discussion of bias in sampling, specifically related to ESM see [Hektner].

Moments of Interest Can also Be Moments for Scripted Action

Above, we described how to define a moment to signal (schedule trigger or event trigger) and how to design a survey. Paco supports more actions than just prompting the user to participate in a study. It also has a full programming library. It allows the specification of more complex behaviors to be carried out as part of a study.

For instance, when an important moment occurs, Paco, can execute a program that evaluates more complicated state and, predicated on that evaluation, the researcher may record sensor data, prompt the user, modify the instrument itself (perhaps changing the sampling frequency), or just postpone action until a more appropriate moment.

Defining Custom Collection Instruments for Surveys

It is also possible to create completely custom surveys using html, javascript and css thanks to Paco's built-in Javascript functions. One example of a custom instrument is a reaction timer interface that allows collecting reaction times for participants.

Deployment and Monitoring

Paco makes it easy to specify who can edit an experiment, who can see the data, what parts they can see, and who can join an experiment.

It also makes it easy to see how participants are engaging on a daily basis with statistics on signaling, response rates and data collection.

Data Export

Paco provides report generation facilities for exporting all data including photos and audio samples in multiple formats including comma-separated value (CSV), Javascript Object Notation (JSON) and HTML.

Running Studies Remotely

Because the research questions that ethnographer and behavioral scientists want to ask are usually best asked in the natural setting of the participant at relevant moments, Paco, works completely offline once the participant has joined the study on the phone. It signals the participant and collects data without requiring any network connections. Experience has shown us that participants usually have systematic failures in network connectivity which leads to systematic sampling errors, e.g., they never have access at home. The data gets uploaded when the phone re-enters a network connected location. If the user has a connection when data is collected, it will be uploaded within milliseconds and available for inspection by the researcher almost immediately.

REIFYING THE RESEARCH INSTRUMENTS AND THEIR COMPONENTS AS FIRST-CLASS COMPUTATIONAL ARTIFACTS

When the user uses the visual designer to build an experiment, the tool is actually building a textual specification document behind the scenes. This document is a new computer program that tells the phones how to run the experiment. This is the basis for the larger goal of, Paco, the platform, experimenting with how to scale the generation of new behavioral instruments and new knowledge in the science itself. Paco is implemented as a construction kit for evolving research components and designs open for all to use and to modify.

This design approach enables computational thinking [Wing] about both qualitative and quantitative behavioral research components. It makes it possible to generate an infinite combination of research instruments from basic building blocks. Specifically, it facilitates this by borrowing practices from three areas of computer science: programming languages, software architecture, and the open source software community.

Specification Language

The research instruments are written in a specification language that is precise and executable. For example,

```
{ "title": "Mobile App Usage Study",
  "creator": “bob@pacoapp.com”,
  "groups": [{"name": “Daily Survey",
          "actionTriggers": [{
          "type": "scheduleTrigger",
          "actions": [{ "actionCode": 1,
                    "type": "pacoNotificationAction",
                    “timeout": 15,  //minutes
                    "msgText": "Time to participate” }],
          "schedules": [{ "scheduleType": 4, // ESM
                     "esmFrequency": 8,
                     "esmPeriodInDays": 0, // daily
                     "esmStartHour": 28800000,   // 8am in milliseconds
                     "esmEndHour": 72000000,   // 8pm in milliseconds
                     "esmWeekends": true
                    }]}],
          "inputs": [{ "name": "question1",
                    "responseType": "list",
                    "text": "Have you used an app since the last time you were
                          signaled?",
                    "multiselect": false,
                    "listChoices": [ “Yes”,”No”] } ],
          "feedback": { "text": "Thanks for Participating!” }]}
```

This specification creates an experiment named “Mobile App Usage Study” that has one task, a survey, named “Daily Survey”. It uses an Experiential Sampling schedule to trigger the participant to participate in the survey 8 times per day between 8am and 8pm. When the user responds, it asks them one yes/no list choice question, “Have you used an app since the last time you were signaled?”. After they respond, it says “Thank you for participating!”.

Paco allows researchers to define many of these types of experiments with the visual builder. This makes it a lot less tedious. It then runs the specifications on the mobile phone or web in a precise, repeatable manner. However, for full power, the researcher can use the programmable building blocks to craft a more precise research instrument by programming directly in a text editor.

While the above specification is a trivial example, it already provides many benefits. Thanks to its precise format, meta-cohort studies become possible. Replication becomes possible. Exact sharing of sub-scales and sampling methods becomes possible.

Most importantly, by defining the what and how of a research instrument in this way, it is open for inspection. The knowledge contained in a research instrument, represented in this precise, open manner facilitates reflection and augmentation and allows new work and new knowledge to be build on it.

If one researcher specifies a research instrument, another researcher can run that instrument. As an example of the value, imagine that the second researcher gets a different outcome from the first. This is almost guaranteed to happen at the current precision of specifying experiments. We refer to this humorously as the, “Confounding Variable Generator.” The different outcome may mean that the specification is not precise enough to be replicable. This is an opportunity to augment the specification so that it contains more detail and precision about the research protocol to be executed. This becomes knowledge about how research is conducted that grows over time and allows ever further improvement.

Just as the invention of writing moved history and knowledge from the oral tradition and made it more transmissible thus giving those societies an advantage, specifying research protocols explicitly extends behavioral sciences’ abilities to build new knowledge more quickly.

The specification declares the "nouns" of the research instrument. Paco also provides access to the "verbs" of research by providing a programming library, in the Javascript language. By making the elements of research design first-class objects, they become directly manipulable by the researcher. They can be composed, extended, and reused. The mechanism inherent in the JavaScript language facilitates abstraction and higher-order reasoning. Abstraction and higher-order reasoning are generative and allow an infinite number of study designs to be expressed. This scales research because research instruments can be constructed that are better at collecting the data needed for the research question at hand.

A Brief Programming Language Explanation

As mentioned above, a programming language defines not only the primitive nouns and verbs but also the means of abstraction and the means of composition [Abelson]. Through those mechanisms, it becomes possible to generate an infinite number of computer programs and raise the level of abstraction, bringing it closer to the problem domain.

Behavior science in this case. Some of the abstractions represent the underlying machine. Some are about ways to compose abstractions into higher-level functions and components.

For example, a computer language may define primitives values, “nouns”, such as a list of word strings,

words = [“the”, “quick”, “brown”, “fox”]
more_words = [“jumped”, “over”, “the”, “lazy”, “dog”]

that represent an abstraction of a word phrase. It may also provide means of composition, such as the ‘+’ operator that allows combining lists into one list:

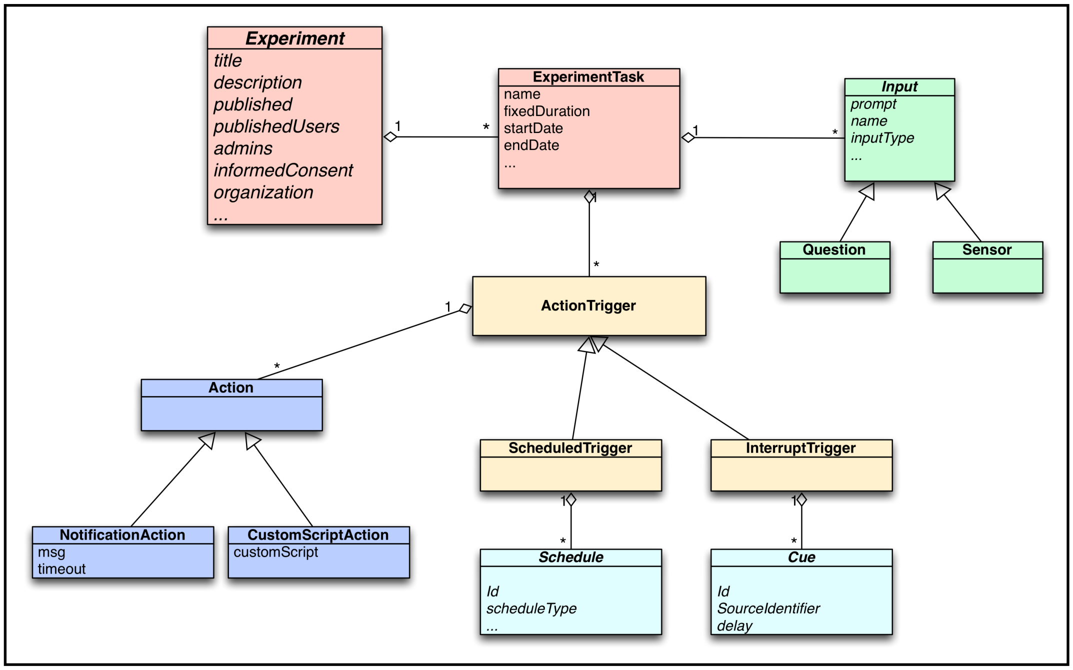
full_sentence = words **+** more_words

Now, thanks to the ‘+’ operator, the variable full_sentence has the value,

[“the”, “quick”, “brown”, “fox”, “jumped”, “over”, “the”, “lazy”, “dog”].

Paco takes this idea and applies it to behavioral science components. Triggering and scheduling definitions, types of data collected, case-based reasoning, sensors, monitoring, visualization, all become components that can be combined into sampling tasks and larger, more specific instruments.

GENERATING NEW INSTRUMENTS AND NEW KNOWLEDGE



In addition to the declarative language for primitives, such as schedules and triggers, described above, Paco uses Javascript, a common general purpose web language, as a base language in which to provide a programming library for building much more advanced experimental instruments. This Paco programming library provides functions, "verbs", for saving and retrieving data in the database, modifying the experiment specification, creating new user interface elements, new compound triggers and new communications to participants in experiments. It also provides functions on android to interact with the system sensors, other applications, and the network. Above all this, JavaScript provides ways to create new functions from scratch or using the already provided functions as building blocks.

More functions are being developed as we speak. Many research groups have started to build higher-level constructs, such as goal setting interventions.

Programming Library Example

Here is an example for programmatically recording data into the data set for an experiment.

```
paco.db.saveEvent({ "experimentName" : "Mobile App Usage Study",
                    "scheduledTime" : "2016/07/22 13:45:01-0800",
                    "responseTime" : "2016/07/22 13:46:15-0800",
                    "who" : "bob@pacoapp.com",
                    "responses" : {  "question1" : "Chrome" });
```

The paco library has a database object, called db, that provides a function, saveEvent, that allows a researcher to enter arbitrary data into the data set collected for a given experiment.

Here is an example for examining the collected data:

```
var events = paco.db.getAllEvents();
var question1Answers = paco.db.getAnswersForQuestion("appUsedQuestion");
```

Once again, the paco.db object is providing a function, getAllEvents, to retrieve all data gathered in the experiment and another function to retrieve the response to one question in a survey, getAnswersForQuestion(*questionName*).

These functions could be used to build visualizations or as a predicate to evaluate in case-based reasoning. For instance, suppose the first time the participant reported using the Chrome browser app, we want to ask for the "Grand Tour" as it is called in ethnography the next time they use it. We could write a program to modify the experiment to do this. We will need a function to test whether it is the first time they responded that they used Chrome.

```
function isFirstChromeUsage(question1Answers) {
  var chromeBrowserUsage = 0;
  for (var answer in question1Answers) {
    if (answer.equals("Chrome")) {
      chromeBrowserUsage = chromeBrowserUsage + 1;
    }
  }
  return chromeBrowserUsage == 1;
}
```

This function looks at all the answers to the *appUsedQuestion* and if the number of responses is 1 then it is the first time that they have used Chrome and the function returns true otherwise it returns false.

We will also need a function to modify the experiment to ask the grand tour questions. Note: we pass the *appname* as a parameter so that we can ask the grand tour for any app we desire.

```
function addGrandTourQuestionAndTriggerForApp(appname) {
  var experiment = paco.experimentService.getExperiment()
  var grandTourQuestion = { "name" : "grandTourQuestion",
                        "inputType" : "opentext",
                       "text" : "I would like to understand what a session using" +
                                   " the " + appname + " app is like. " +
                              "Can you describe how you typically use it from " +
                              "start to finish?",
                       }
 experiment.groups[0].inputs = [ grandTourQuestion ];

 var appUsageTrigger = { "type": "interruptTrigger"
                        "cues": [{"cueCode": 5, // "app stopped" trigger
                        "cueSource": appname }],
                        "actions": [{"type": "pacoNotificationAction",
                                  "timeout": 15,
                                  "actionCode": 1}],
                        "minimumBuffer": 59
                       }

 experiment.groups[0].triggers.add(appUsageTrigger);
 paco.experimentService.saveExperiment(experiment);
}
```

This function rewrites the survey questions in the experiment to ask them to describe their usage of the given app, *appname*, from start to finish, in the manner of a grand tour question.

Let’s put these two functions together now. When the user responds and hits the save button, the following functions evaluates the situation using the first function and if true adjusts the experiment using our second function.

```
function saveResponses(formResponses) {
  paco.db.saveEvent(formResponses)
  var events = paco.db.getAllEvents()
  var question1Answers = paco.db.getAnswersForQuestion(“appUsedQuestion”)
   if (isFirstChromeUsage(question1Answers)) {
     addGrandTourQuestionAndTriggerForApp(“Chrome”)
   }
}
```

By making the addGrandTourQuestion function take the app name, we can now easily construct a grand tour question for any app we are interested in, not just the Chrome App, and reuse that in a new experiment.

This is just one trivial, contrived, example demonstrating how to use the Paco programming library to dynamically adapt a set of interview questions in a fielded experiment. Most examples are much more complex than space allows us to explore here.

PRINCIPLES FROM SOFTWARE ENGINEERING AND OPEN SOURCE

From software engineering, Paco borrows the best practice of modular architecture and system design to allow extensibility and the practice of automated unit testing for quality and regression testing.

With a modular architecture comprised of components, it becomes easier to build in new sensors and new analytical tools to support the relentless development in hardware and research methods. More sensors are being developed as we speak. External researchers have contributed new sensors such as calendaring integrations, fitness sensor integrations and context awareness sensors such as location.

Unit testing is a programming practice that checks the implementation of computer functions at a fine grained level. It helps ensure that a function behaves as intended and in an ongoing software project it serves as a regression test that catches unintended program breakage due to the deletion and modification of existing code and the addition of new code. This safety net makes it easier to have many parties contributing code to the project and to ensure that the project minimizes bugs.

Open Source Is Open Science

From the software community the Paco project adopts the philosophy that problems as large as behavioral science require a community of researchers and developers working together openly. Therefore, it is licensed as open source software, uses open tools for development [Github], and invites collaborators in industry and academia. This open-ness means that any researcher can build on what others have done. They can examine the implementation to see exactly how it works and they can build new features into it to accommodate their particular needs as well as provide a clear, common description of their own research.

Another reason it needs to be open is that the research questions to be asked are infinite in number and always evolving. The effort benefits from what Eric Raymond, famed UNIX hacker, calls a "network effect."[Raymond] Behavior is far too vast a domain for any single group of people to possibly be able to cover any significant part of it in any useful time frame. It requires a network, a community, of researchers applying and improving the methods if it is to make significant progress.

Currently, every computer tool used for research reinvents the wheel and usually in a proprietary, non-extensible way. Research has to bend to the tools and the variability of commercial enterprises. Notably, commercial interests are much more "factist" oriented rather than "interpretivist" oriented. This leads to tools that help companies make money by providing features that support the largest number of researchers rather than helping individual researchers dig deeply into their unique questions.

This is one more reason why it needs to be open source so that it can continue to grow beyond any one company's lifetime. This is an ongoing scientific endeavor that will take may take decades or longer.

In open source software, an analogous system is Linux, the operating system that runs 54% of mobile devices and 96% of servers on the internet [Wikipedia] of the data centers and phones in the world. As new hardware comes along, as new ideas about efficient system design come along, as new applications come along, interested parties can add to the system, modify the system, or "fork" the system and make changes based on their advances and needs.

Many of these changes are given back to the system. This sharing of advances makes the system better for everyone while distributing the work load.

LIMITS AND CHALLENGES

There are many limits and challenges to our project. Some of the most important challenges are in the very heart of this endeavor, whether we can automate any aspects of qualitative, descriptive inquiry in a way that improves scale and still maintains quality. We examine the problems that might occur if this were done within a positivist framework. Not least of the problems is the deep knowledge of humanity required to conduct qualitative research. We also have to consider what is computable. After that, we have to deal with the problem of ensuring that our methods are created to do what we actual intend them to do. Are the implementations, assuming multiple platforms, compliant with our specification. Even if we can implement some subset of methods correctly, there are limits on whether they will be adoptable by researchers.

Positivism Is No Panacea Nor Is It Our Proposed Program

It is important to clarify that this proposed path is no panacea and that it is not simply positivism in high-tech clothing. Making a research protocol more explicit is just a way of saying what we actually are doing clearly so that others can more likely replicate the findings, or not. That replicability adds to the knowledge base by instilling more or less confidence in the protocol's outcomes. In many sciences recently there has been a huge crisis in replicability [Baker], there are many reasons, but 55% of scientists surveyed by Nature cited "Methods, code unavailable".[Ioannides]

A problem of behavioral science, due to the humans and human societies involved, is that it is so much more complex than the "hard" sciences. A program to precisely define research more clearly might unify concepts, clearing out duplication and pointing out uncharted territory.

Another feature of positivism, not required by this proposal, is the need to frame a research hypothesis in a form that is falsifiable. This is not the pursuit of ethnographic research nor of our project to make experiments that are more precisely defined. The pursuit is to understand culture and to use that description to illuminate the human experience whether it be to design products that are more useful or to design interventions that make life more healthy and happy.

The description should be precise enough that the results gathered using a method conforming to it conveys the participant's perception of reality repeatably and faithfully.

This description, being language-based, and having meaning imparted to it by the researchers using it and the participants they study using it is subject to seemingly irreducible conflicts. Human language is not precise the way computer languages are precise. Ask anyone who ever tried to use a "human-language" programming language such as AppleScript [Konwinski]. Another example of the ambiguity can be seen in references catalogued by Winograd schemas [Winograd]. For example:

> The city councilmen refused the demonstrators a permit because they *feared* violence.
>
> The city councilmen refused the demonstrators a permit because they *advocated* violence.

Even in a specialized field like ethnography, different schools may use the same word differently. A specification language must either be more constrained than human language in its definitions or it must be more verbose if it is to be precise.

Behavioral Techniques and Limits on What is Computable

Another dimension in which we are currently limited is in specifying methods that are outside of the domain of the tool. As an example from the Agile Science movement [Agile], a coaching intervention is a human interaction where a "coach" guides a participant through a behavior management protocol. The specification of this intervention is conceivable if it is an artificial intelligent agent because that means it is inherently limited in abilities. If it is a human being then there is a seemingly infinite degree of variation in how the coach will be specified. It may be a strict disciplinarian coach. It may be a nurturing, accepting coach. It may blend those properties in at different circumstances driven by complex cultural norms deployed in specific contexts. It may be impossible to precisely specify such interactions because of the sheer enormity of variables and the complexity of the behaviors. Or, it may be possible. Only experimentation will tell.

This directly reflects a challenge in the attempts to conduct ethnographic interviews with computational methods and one of the reasons that computational tools are augmentations of qualitative methods not replacements. Interviewing, the hallmark of ethnographic field research, is a process that is as complex as the richest human activity. One example of the complexity is the deep reading required prior to entry into the field. That knowledge is very helpful if the researcher is to understand what is significant in the speech acts of the participant and what should be explored further. That deep knowledge trains their research eyes.

Ethnography also requires the researcher to be able to process cultural differences, as expressed by other humans, in their own cultural terms. This means quick interpretation of new symbols or symbols used in new ways from limited numbers of examples. Also, the input is coming from a participant who may or may not clearly understand what they themselves deem to be significant. Good interviews, according to Ladner [Ladner] are heavily guided by theory and by an intuitive sense that comes from being an experienced interviewer who can build rapport.

In contrast, there has been a lot of promise building rapport with even very simple chat programs such as Eliza [Weizenbaum] and there is currently a re-surging interest in ai-driven bots in communications apps [Edwards].

At the end of the day though, it is very hard to automate processes that we can't describe clearly enough to program.

There is another deeper limit of what is computable, Godel's Incompleteness Theorem, which sets limits on what is decidable in a logical system but it is beyond the scope of this article to address and the project is not likely to bump into it for a very long time to come.

Compliance of Execution

Given that the list of variables that might contribute to a clear specification is infinite, we have to expect boundaries on what this project could accomplish. Granting that, there are other obstacles that confront the project as well. One is the faithful execution of the specification across platforms and across implementations.

Any project this large will likely have multiple competing implementations. If the research generated across implementations is to be useful, it must be comparable. The expected outcomes must be testable. It must be possible to verify that the same protocol produces the same outcomes. This will be quite difficult given that there are so many other unspecified variables in any given execution of the protocol that thwart repeatability at least in the beginning. Even with the same participants there might be changes. They will have had experiences since the previous execution in ways that may alter the outcome. Most obviously, the experience of participating may itself have caused reactivity that alters outcomes.

This same problem applies even within the same implementation because of other changes such as defective hardware, upgraded operating system software, and a host of other variables inherent in complex general purpose computing systems.

The process of implementation must be coupled with a process for verifying outcomes if the specifications and the data they generate are to contribute to the advancement of our knowledge of behavior and culture.

Computational Thinking in Behavioral Science

Lastly, one other challenge to consider is the ability to capitalize on the advances presented here are limited by the ability to think computationally [Wing]. To make research items first-class abstractions, to compose compound instruments out of them, to decompose a research design programmatically requires the ability to think algorithmically. For example, imagine wanting to collect some data after observing a phenomenon occur several times. This is simple case-based reasoning that we do intuitively in our heads. It is another set of skills to tell a computer to do it. The researcher must write a recipe for the computer. First it makes sure the computer senses and records instances of the phenomenon. Then it must make sure that the computer periodically reviews the collecgted observations to see if a number of occurrences has been observed. This requires storing the data and setting a schedule for evaluating that data at the appropriate intervals. This decomposition of a problem into a set of exact steps is a skill many are not yet used to exercising. The hope is that computational thinking will become core curriculum in primary and secondary education at some point but it is not yet. In the meantime, exercising this skill in an advanced way may require collaboration between disciplines.

SUMMARY

Paco has already helped hundreds of researchers build research instruments quickly that are more precisely suited to their research questions and have allowed them to scale data collection. By incorporating computational tools into the platform that reify the research instruments and methods, Paco can help many more researchers build new, more precise instruments and hopefully hasten the generation of new knowledge in the behavioral sciences.

Our overall approach can be best summarized with the following quote from an eminent computer scientist.

> Expressing methodology in a computer language forces it to be unambiguous and computationally effective.
>
> The task of formulating a method as a computer-executable program and debugging that program is a powerful exercise in the learning process.
>
> The programmer expresses his/her poorly understood or sloppily formulated idea in a precise way, so that it becomes clear what is poorly understood or sloppily formulated.
>
> Also, once formalized procedurally, a mathematical idea becomes a tool that can be used directly to compute results.
>
> —*Gerry Sussman, Professor, MIT*

By building and deploying research instruments, in a computational manner, we can reflect on those instruments, refine them, improve our knowledge of behavior, and build ever better instruments.

Bob Evans is a toolmaker and computer scientist at Google. His work is dedicated to augmenting human intelligence and quality of life by providing tools to support analysis and exploration of daily experience. Bob is the creator of PACO, an open-source, mobile, behavioral science research platform. He previously worked at Fujitsu, Borland Software, and Agitar Software.

NOTE

The ideas presented herein are mine alone and do not represent the position or opinions of my employer in any way.

APPENDIX A - FULL EXPERIMENT SPECIFICATION EXAMPLES

Experiment 1- An ESM study

```
{
 "title": "Mobile App Usage Study",
 "creator": “bob@pacoapp.com",
 "contactEmail": "bob@pacoapp.com",
 "id": 0000,
 "recordPhoneDetails": false,
 "extraDataCollectionDeclarations": [],
 "deleted": false,
 "modifyDate": "2016/07/26",
 "published": false,
 "admins": [
  "rbe5000@gmail.com"
 ],
 "publishedUsers": [],
 "version": 3,
 "groups": [
  {
   "name": “Daily Survey",
   "customRendering": false,
   "fixedDuration": false,
   "logActions": false,
   "logShutdown": false,
   "backgroundListen": false,
   "actionTriggers": [
    {
     "type": "scheduleTrigger",
     "actions": [
      {
       "actionCode": 1,
       "id": 0000,
       "type": "pacoNotificationAction",
        "snoozeCount": 0,
        "snoozeTime": 600000,
        "timeout": 15,
        "delay": 0,
        "color": 0,
        "dismissible": true,
        "msgText": "Time to participate",
        "snoozeTimeInMinutes": 10
      }
    ],
    "id": 0000,
    "schedules": [
      {
        "scheduleType": 4,
        "esmFrequency": 8,
        "esmPeriodInDays": 0,
        "esmStartHour": 28800000,
        "esmEndHour": 72000000,
        "signalTimes": [
          {
            "type": 0,
            "fixedTimeMillisFromMidnight": 43200000,
            "missedBasisBehavior": 1
          }
        ],
        "repeatRate": 1,
        "weekDaysScheduled": 0,
        "nthOfMonth": 1,
        "byDayOfMonth": true,
        "dayOfMonth": 1,
        "esmWeekends": true,
        "minimumBuffer": 59,
        "joinDateMillis": 0,
        "id": 0000,
        "onlyEditableOnJoin": false,
        "userEditable": true
      }
    ]
  }
],
"inputs": [
  {
    "name": "question1",
    "required": false,
    "conditional": false,
    "responseType": "list",
    "text": "Have you used an app since the last time you were signaled?",
    "likertSteps": 5,
    "multiselect": false,
    "numeric": false,
    "invisible": false
```

```
          "listChoices": [
            "Yes",
            "No"
          ]
        }
      ],
      "endOfDayGroup": false,
      "feedback": {
        "text": "Thanks for Participating!",
        "type": 0
      },
      "feedbackType": 0
    }
  ],
  "ringtoneUri": "/assets/ringtone/Paco Bark",
  "postInstallInstructions": "<b>You have successfully joined the
                          experiment!</b><br/><br/>\nNo need to do anything else for
                          now.<br/><br/>\nPaco will send you a notification when it is time to
                          participate.<br/><br/>\nBe sure your ringer/buzzer is on so you will hear the
                          notification."
}
```

REFERENCES CITED

Abelson, H. and Sussman, G.J. (1996). Structure and Interpretation of Computer Programs (2nd ed.). Cambridge, MA: MIT Press.

Agile Science Website, July 2016. http://www.agilescience.org.

Baker, M. "1,500 scientists lift the lid on reproducibility". Nature, July 2016. http://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970?WT.mc_id=TWT_NatureNews%5D

Christensen, C. M. (1997). The innovator's dilemma: When new technologies cause great firms to fail. Boston, Mass: Harvard Business School Press.

Christian, B., & Griffiths, T. (2016). Algorithms to live by: The computer science of human decisions. New York, NY: Henry Holt and Co.

Edwards, C. "Why and how chatbots will dominate social media." TechCrunch, July 2016. https://techcrunch.com/2016/07/20/why-and-how-chatbots-will-dominate-social-media/.

Github repository for Paco: https://github.com/google/paco.

Glaser, B.G. & Strauss, A.L. (1967). Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago,: Aldine.

Hektner, J. M., Schmidt, J. A., & Csikszentmihalyi, M. (2007). Experience sampling method: Measuring the quality of everyday life. Thousand Oaks, Calif: Sage Publications.

Ioannidis JPA (2005) Why Most Published Research Findings Are False. PLoS Med 2(8): e124. doi: 10.1371/journal.pmed.0020124

Konwinski, A. (2007) "Applescript, the most unnatural natural language." Personal Blog, July 2016. http://andykonwinski.com/applescript-the-most-unnatural-natural-language/.

Ladner, S. (2014) Practical Ethnography: A Guide to Doing Ethnography in The Private Sector. Thousand Oaks, California: Leftcoast Press.

LeComte, M.D. & Goetz, J. (1982) Problems of Reliability and validity in ethnographic research. Review of Educational Research, 52, 31-60.

Paco Website: https://www.pacoapp.com/

Phenomenology, Stanford Encyclopedia of Philosophy, July 2016. http://plato.stanford.edu/entries/phenomenology/

Raymond, E. S. The Cathedral and the Bazaar, July 2016. http://www.catb.org/esr/writings/cathedral-bazaar/

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (January 1966), 36-45. DOI=http://dx.doi.org/10.1145/365153.365168

Wikipedia. Usage Share of Operating Systems. Wikipedia, July 2016. https://en.wikipedia.org/wiki/Usage_share_of_operating_systems.

Wing, J.M. 2006. Computational thinking. *Commun. ACM* 49, 3 (March 2006), 33-35. DOI=http://dx.doi.org/10.1145/1118178.1118215

Winograd, T. 1972. Understanding Natural Language. New York: Academic Press.