Lowering the Barrier for Crowdsensing Application Development

Scott Heggen, Amul Adagale, and Jamie Payton

Department of Computer Science
The University of North Carolina at Charlotte
{sheggen,aadagale,payton}@uncc.edu

Abstract. Crowdsensing has the potential to support human-driven sensing and data collection at an unprecedented scale. While many organizers of data collection campaigns may have extensive domain knowledge, they do not necessarily have the skills required to develop robust software for crowdsensing. In this paper, we present Mobile Campaign Designer, a tool that simplifies the creation of mobile crowdsensing applications. Using Mobile Campaign Designer, an organizer is able to define parameters about their crowdsensing campaign, and the tool generates the source code and an executable for a tailored mobile application that embodies the current best practices in crowdsensing. An evaluation of the tool shows that users at all levels of technical expertise are capable of creating a crowdsensing application in an average of five minutes, and the generated applications are comparable in quality to existing crowdsensing applications.

Keywords: crowdsensing, participatory sensing, mobile phone sensing, end-user programming.

1 Introduction

Crowdsensing provides an opportunity for a fundamental shift in the way governments, organizations, research institutions, communities, and individuals gather data to make decisions. In this emerging class of software systems, participants use the sensors (e.g., cameras, GPS, accelerometers) and input capabilities of their smartphones to collect digital samples of the surrounding world for a data collection campaign, typically organized to address a scientific question or civic issue. Such an approach can supplement data from special-purpose sensors, or even replace their use, providing data from a fine-grained, human perspective and potentially reducing the costs of large-scale data collection efforts.

Crowdsensing is driven by volunteers who use the sensors embedded in their smartphones to collect data. Crowdsensing applications have been developed for a variety of domains, including environmental monitoring [21,25,18], wildlife and habitat monitoring [4,20,29], health and well-being [33,6,15], social networking [22], road traffic monitoring [37], and fuel-efficient driving [7]. Given the

G. Memmi and U. Blanke (Eds.): MobiCASE 2013, LNICST 130, pp. 1–18, 2014.

success of these initial deployments and the commercial success of crowdsourcing (e.g., [2]), it is reasonable to expect a significant increase in demand for crowdsensing systems.

Campaign organizers (i.e., individuals interested in distributing a crowdsensing application as a data collection tool) may include scientists, community organizers, interested citizens, and hobbyists. Given the diverse nature of potential campaign organizers' perspectives and areas of expertise, it is likely that most will lack the skills or resources to develop such systems. Our goal is to empower campaign organizers, with limited or no software development experience, to create their own crowdsensing applications.

To this end, we present Mobile Campaign Designer (MC Designer), an end-user programming tool for creating and managing crowdsensing campaigns. MC Designer provides a simple interface for campaign organizers to define characteristics of their crowdsensing campaign and generates a tailored mobile application. This generated mobile application is capable of collecting and submitting data using sensors commonly embedded in smartphones. To demonstrate that MC Designer makes crowdsensing application development more accessible, we conduct a small user study. First, we evaluate the user's ability to use MC Designer to create a crowdsensing application for a given scenario. Second, users compare an existing crowdsensing application to one created using MC Designer. Our results indicate that users can quickly and easily create applications with MC Designer that are comparable in quality to existing crowdsensing applications.

2 Related Work

Crowdsensing stems from public participation in scientific research, or citizen science, in which volunteers collect data for a scientific purpose. Citizen science's long history began with the National Audubon Society's annual Christmas Bird Count (CBC) [27], an extremely successful data collection campaign that highlights the potential for citizen science. Over 63,000 volunteers submitted almost 65 million observations in the most recent CBC [19]. The results of the society's 112 years of citizen science has been used in 300 publications.

Crowdsensing extends citizen science by incorporating mobile phones, providing volunteers with sensors for sampling data. Crowdsensing has shown promise for studying urban environments [25,6,7,34,24], social interaction [22,8], health and wellness [33,12], education [9,11], and biology [4,20,28]. In order for crowdsensing to become widely available, however, mobile application development needs to be simplified for non-technical campaign organizers. Many campaign organizers will have domain expertise but lack the skills to create an industrial-grade mobile application to support data collection, and may often lack the resources to contract a professional software developer.

Some tools exist that aim to simplify application development, such as GUI-based and end-user programming systems for mobile operating systems. For example, MIT's App Inventor [23] provides simple GUI-based interactions for creating mobile applications for the Google Android mobile operating system.

Microsoft's TouchDevelop [38] takes end-user programming one step further by putting the development environment on the mobile phone. However, both require a basic understanding of programming concepts and syntax to build any application. Also, both systems are general-purpose application-creation tools. Crowdsensing does not require most of these features (such as sprites for gaming), which distract from the goals of campaign organizers.

Close in spirit to our work are approaches specific to crowdsensing that attempt to eliminate the need for programming skills to create data collection campaigns. For example, Project Noah [29] is a tool for creating "missions" in which users can contribute images of wildlife. Campaign organizers register a mission on the website, and users simply upload their images from any camera via Project Noah's website. Similarly, Epicollect [1] allows the creation of campaigns specific to epidemiology and ecology. Participants collect data using the Epicollect mobile application, which provides an interface for collecting camera, GPS, and text-based data. While both Project Noah and Epicollect eliminate the need for developer skills, these systems are specific to their respective domains and do not provide general support for crowdsensing. Sensr [17] provides more generalized support by providing campaign organizers with a web interface for creating a campaign, which users then access via the Sensr mobile application. However, Sensr is limited to camera data and text-entry input. All three of these tools fail to provide access to many desirable sensors, including the accelerometer, camcorder, and microphone.

More sophisticated crowdsensing campaign creation tools have been explored as well, such as Campaignr [14], Medusa [30], PRISM [5], Code-in-the-Air (CITA) [31], and the Open Data Kit (ODK) [10]. Each of these provide high-configurability of campaigns by the end user, remove the user from the technical challenges of accessing phone sensors, and are robust to changes in the campaign. However, they still require some programming knowledge, including knowledge about the format and ordering of a document, correct use of the programming language's syntax, or infrastructure knowledge. In many cases, this may still present too large of a barrier for users. MC Designer was specifically designed to lower the barrier for the campaign organizer by providing an easy-to-use graphical user interface that allows users to create a mobile application simply by supplying relevant crowdsensing campaign parameters; MC Designer handles all configuration tasks based on the campaign organizer's description of the campaign. Furthermore, many of these systems require participants to use a single mobile application for all campaigns. MC Designer generates a stand-alone mobile application tailored specifically for each campaign. Lastly, MC Designer runs on mobile phones, allowing campaign organizers to quickly build campaigns from their smartphones in situ.

3 The MC Designer System Architecture

Our goal is to support the widespread adoption of crowdsensing by providing a tool that allows users to easily create a crowdsensing application. In this section, we describe the desired features that motivate our design choices. We then present the system architecture of MC Designer.

3.1 MC Designer Features and Design Motivation

MC Designer has two primary stakeholders: the campaign organizer and the campaign participant. The campaign organizer's goals are two-fold: gather data reported by participants using their mobile device and analyze the data to draw meaningful conclusions. The campaign participants' goals are to use their mobile device to contribute data to the campaign.

To meet these goals, MC Designer provides the campaign organizer with a mobile tool for defining all the parameters of a crowdsensing campaign, which are then used to generate a tailored mobile application serving the specific purposes of the campaign. Second, a campaign organizer needs participants to collect data; MC Designer addresses recruitment by providing the campaign organizer with a means to invite known parties or to recruit participants from a pool of identified participants through a *profile matching* system. The profile matching system allows interested volunteers to provide a profile that describes their interests and personal characteristics. These profiles are matched to new campaigns, and prospective participants are invited to join via email. Currently, the profile matching system allows users to be invited to campaigns based on an age range, gender, ethnicity, and geographic location. Future work will incorporate more sophisticated profile matching systems that evaluate potential partipants' trustworthiness based on their previous contributions, like those introduced in [32,13]. Once the crowdsensing application has been generated and potential participants have been recruited, the application is distributed to potential participants, who can use the app to collect and submit data samples.

Lastly, the campaign organizer must be able to access the collected data. The campaign organizer has two options: explore the data through a web interface dedicated to their campaign, or download the entire data set for offline analysis. Access to either option is available through the web interface on the MC Designer website.

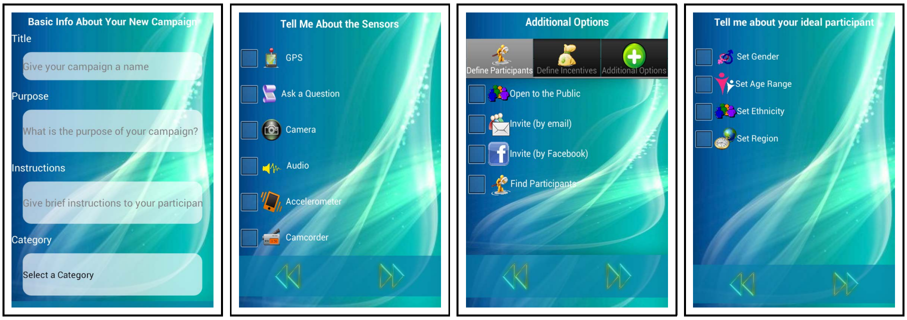
Taking these goals and challenges into consideration, we designed MC Designer with five major subsystems:

1. **Mobile Campaign Manager:** interface for creating a crowdsensing campaign
2. **Application Server:** generates the tailored crowdsensing application
3. **Data Management Server:** receives data submissions from participants
4. **Web Interface:** interface for campaign organizers to access their data
5. **Campaign Application:** mobile application created by campaign organizer for collecting data

3.2 Mobile Campaign Manager

A campaign organizer uses the Mobile Campaign Manager (MCM) mobile client to define the parameters of their crowdsensing campaign. We chose to design

MCM as a mobile application for two reasons. First, mobile phones are quickly becoming the primary means of computing for many people thanks to their increased computing power, small form factor, and constant connectivity to the internet. Second, with this approach, the campaign organizer can create the mobile application in situ, instead of having to leave the field, facilitating on-the-fly campaign creation.



(a) (b) (c) (d)

Fig. 1. When defining a new campaign, the user configures relevant parameters in MCM: (a) basic information, (b), sensors, (c) participants, (d) profile matching for recruiting participants (optional), and additional options (optional, not shown). The user can review their campaign before submitting to the server.

Figure 1 shows the activities performed when creating a campaign. One of the most significant decisions the campaign organizer makes is the selection of sensors (Figure 1b). Because the energy consumption associated with a crowdsensing app is a concern, MCM allows the campaign organizer to configure the sampling rate and other characteristics of data capture quality. Currently, MCM supports adjusting the polling frequency of the GPS receiver when selected by the campaign organizer. However, since the campaign applications are currently created for the Android OS, application-level adjustment of the sampling rate is limited for most of the other sensors. The Android OS optimizes the sampling rate of embedded sensors and allows programmers only to specify the polling rate of associated sensor listeners. In other words, Android only enables the sensor when a listener is registered to the sensor; the sensor operates at the frequency dictated by the OS, but only notifies a listener at the listener-specified polling frequency.

When the campaign organizer finishes defining their campaign, MCM generates an XML file that captures the campaign parameters and sends it to the application server, where it will be used to build a tailored application. MCM essentially acts as a centralized campaign management system; in addition to creating an app, the campaign organizer can install and launch existing campaign applications. Currently, MCM can be deployed on phones with Android OS 2.3.3 or newer.

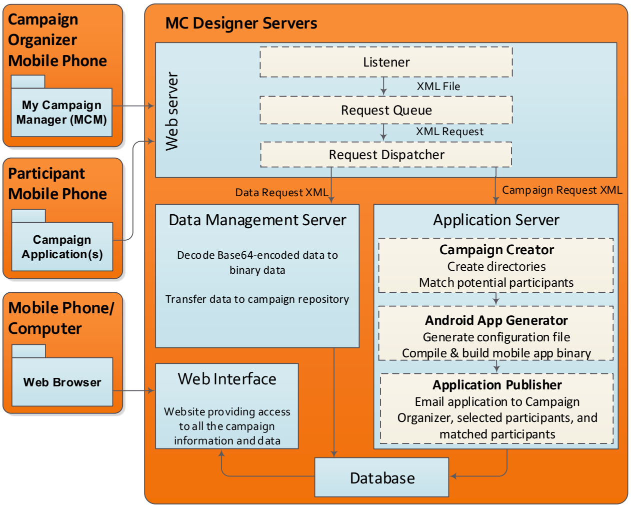


Fig. 2. MC Designer Software Architecture

3.3 The MC Designer Servers

The MC Designer server architecture is shown in Figure 2. First, all requests from MCM and the campaign applications are sent to a *web server* which responds to incoming HTTP requests. As processes become available, these XML requests are dispatched to either the application server or the data management server through the *request dispatcher*.

The Application Server. The application server is responsible for generating the source code and an executable for the crowdsensing application. The *campaign creator* parses the campaign request XML generated and sent from MCM, and creates a storage repository for the application source code and a directory for data submissions to the campaign. Next, a **constants** file is generated, which contains the configuration parameters that determine the user interface and resulting behaviors of the app, such as which sensors are available for use by the user. The constants file also stores important information about the campaign, such as identifiers for the campaign and the user, local storage directories, and HTTP parameters to connect to the server to submit data.

The source code and constants file are used by the *app generator* to build an application binary that can be installed on a mobile device. Each campaign application is given a unique package name, allowing multiple campaign applications to be installed on a single device. Should the campaign organizer want to make the application available via a mobile application market, the source code is made available to them. They are free to modify the code, sign the application, and place it in the market. After the application is created, the *application publisher* notifies the campaign organizer that her campaign is ready, and makes the campaign accessible through her instance of MCM. The application publisher also contacts all invited participants (directly through email or social networks, or via profile matching notifications) to give instructions for installing the application. Lastly, MC Designer organizes all information about campaigns and data submissions in a MySQL database.

The Data Management Server. Data submissions from campaign applications are also transferred to the server as an XML file packaged in an HTTP request. The data management server parses the request to determine the appropriate campaign and user information, decodes the data from the XML file, stores the data in the appropriate directory, and updates the database with the new submission. A campaign can use multiple sensors, and each piece of sensor data may have relationships with other sensor data (e.g., GPS data and an image, so the image can later be placed on a map). Each piece of sensor data is stored on the server as a separate file, but is tagged in the database with a session ID. The session ID is used to maintain the connection between data from a single data capture session so correlations between data points can be discovered during data analysis.

3.4 The Web Interface

MC Designer also consists of a website, which allows the campaign organizer to view and analyze data that has been submitted to the campaign. Each campaign has a web page that presents a campaign description, a link to download the campaign application, and summary information about the data collected (e.g., number of total submissions, coverage of a geographic area). The webpage also provides access to basic analysis tools, such as the ability to plot GPS points on a map and graph accelerometer data. Lastly, campaign organizers have the ability to download the entire data set.

3.5 The Campaign Application

Each time a campaign organizer defines and submits a campaign using MCM, a *campaign application* is created. The campaign application's primary purpose is to allow participants to collect data using the phone's sensors. The campaign organizer has six sensors (and their parameterized options) to chose from when defining their campaign: accelerometer, camcorder, camera, GPS, microphone, and text-entry. The application incorporates the correct combination of sensors and integrates their functionality to avoid conflicts (e.g., passing control of the microphone to a video capture session). Once the user has captured data, the campaign application allows the user to send these sensor readings to the server. To ensure a single interface between the campaign application and the server, all data is encoded into Base64, which converts any binary data into ASCII-based data. The encoded data is then added to an XML file. The use of XML is motivated by its widespread adoption and its ability to encode all data types in Base64, including potential future data types.

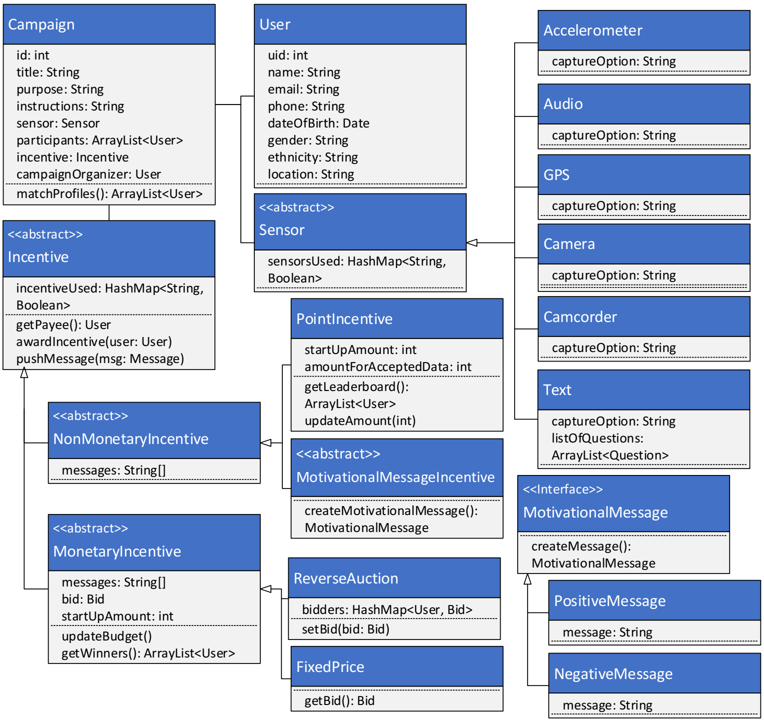


Fig. 3. MC Designer Class Diagram

3.6 Extending MC Designer

MC Designer incorporates many features for building crowdsensing apps and is extensible, with an API that provides hooks for incorporating emerging solutions in crowdsensing research. Figure 3 highlights some of the classes in MC Designer that contribute to this extensibility. For example, the `Sensor` abstract class and its concrete descendants provide access to the low-level physical sensors on the phone, and the `Sensor` class can be extended to provide MC Designer applications with access to higher-level sensors (e.g., utilizing sensor fusion). MC Designer also provides implementations of basic incentives (e.g., fixed price micropayments), and the `Incentive` class can be extended to incorporate more advanced mechanisms. MC Designer is open source and is available via an SVN repository (`https://subversion.assembla.com/svn/PSToolkit/`), allowing researchers to integrate their own implementations of crowdsourcing concepts into the tool.

4 Evaluating MC Designer

The motivation for creating MC Designer was to simplify development of crowdsensing applications. In our evaluation, then, we are concerned with 1) the ability of users with varying degrees of programming experience to develop crowdsensing applications with MC Designer, and 2) the expressiveness of MC Designer. We conduct a two-part user study, in which participants use MC Designer to create crowdsensing applications for a scenario and then evaluate the quality of the created applications as compared to existing implementations of custom-designed crowdsensing applications.

In part one of our study, users were provided with a mobile phone and asked to use MC Designer to create two mobile applications using scenario descriptions to guide their design. After creating the application with MC Designer, the study participant was asked to complete a survey to evaluate the application on two primary criteria: 1) is the application what you expected, and 2) based on your understanding of the scenario, does the application serve the purposes of the scenario?

Specificially, users in the study were presented with two scenarios describing a crowdsensing campaign:

> Scenario A: You are an astronomer interested in collecting data about meteor showers. Using your application, you would like to provide a method for users to provide textual input regarding the conditions during the meteor shower observation, such as sky conditions and visibility of stars (the camera and camcorder are not expected to be used to capture this data). You would also like to let the users record audio annotations as they are observing meteors. Lastly, users need to be able to input the number of times a meteor is spotted.

> Scenario B: You are a physical therapist, and you are constantly concerned about the conditions in which you use your bicycle every day. As such, you would like to build an app that allows users to capture data about their surroundings while riding. You would like to know the exact route taken, as well as the amount of noise encountered and the roughness of the road. Additionally, you'd like users to be able to take pictures or videos of dangerous encounters or other obstacles to bicycling.

These scenario descriptions were derived from real-world existing crowdsensing applications with publicly available implementations. Scenario A describes the Meteor Counter application [26], which was built for NASA to allow users to count meteors during a meteor shower. Users can enter information about the sky conditions, the star visibility, as well as use the microphone to provide audio annotations about each meteor. Scenario B describes Biketastic [34], which uses the GPS to track location, the microphone to record noise, the accelerometer to measure the roughness of the road, and the camera and camcorder to provide information about obstacles and points of interest along the bike route. These applications were selected because they incorporated numerous sensors and they have robust implementations that are publicly available.

In part two of the user study, the native implementations of the applications for both scenarios (Meteor Counter and Biketastic) were given to the user. For each, the user was asked to compare the custom-built implementation to an application generated by the research team using MC Designer. The purpose in providing the user with an app created by the research team was to 1) reduce any bias by the user, since they are the "creator" of the application and may have formed a sense of pride or ownership of the application, 2) ensure the MC Designer-created application mimics the real application as closely as possible, and 3) ensure a consistent evaluation of a single MC Designer-generated application across all users.

A pre-survey was issued to gauge the user's understanding of technology and familiarity with crowdsensing. The user was surveyed after each scenario in part one of the study, to gauge their perception of the campaign creation process. After each scenario in part two, the user was surveyed again to capture their observations while comparing the real application to the one created by MC Designer. Lastly, the user was surveyed at the end of the study to gauge their understanding of the process they'd just completed and their overall perception of the MC Designer system. Throughout the user study, participants were audio-recorded and asked to "think out loud" as they performed the activities. These recordings were reviewed to gain a better understanding of the user's thought process and perceptions of the system while using MC Designer.

5 User Study Results

The user study included 19 participants ranging in age from 18 to 55; 26% were female, 74% were male; all participants were either faculty, staff, administrators, or students at the university. Participants were evaluated on their technical competency prior to the study with a focus on their mobile and smart phone usage, since these are the primary tools used by MC Designer.

All users indicated they owned a mobile phone and used it frequently (more than once a day), and 79% of the users indicated they owned a smartphone. The majority of these users (70%) owned an Android-based smartphone, which is the mobile OS used in our study. When asked to rate their comfort level with software in general, learning to use new software, and their comfort level using a touch screen on a mobile phone, users ranked their comfort levels quite high (8.21, 8.00, and 8.21 out of 10, respectively). Given these observations, it is reasonable to assume that users were not greatly influenced by an unfamiliarity with mobile technology.

Users were also asked to rank their ability to create software using traditional programming languages (e.g., Java, C++, Python) and development tools. Overall, the average rating was 4.42 on a 10-point scale. Four users rated their ability to create software at 1 (no competency), and two users rated 10 (highly competent). The remaining 13 users rated themselves between 3 and 8, with the majority rating a 3, indicating that most users had some basic programming skills but did not feel they were "expert" programmers. We also compared the results excluding the two users who rated 10, and no significant differences in the results were identified. The remainder of this section reports on all 19 users.

5.1 Part I: Creating Applications with MC Designer

Part one of the study was designed to evaluate the user's ability to create a mobile application using MC Designer. Users were given the two scenarios and were asked for each to create a mobile app that served the purposes of the scenario. After creating the application, users explored the campaign application and described how well the application met their expectations. Surveys were issued after each scenario to capture 1) the user's level of confidence in MC Designer's ability to create a campaign application, 2) the actual and perceived accuracy of the application, and 3) the amount of time it took the user to create the application.

Confidence in MC Designer. In scenario A (Meteor Counter), users were confident in MC Designer's ability to generate a mobile application, despite having never used the system before, as shown in Table 1. Aside from question 4, users rated the mobile application they created above average in each category. Users were very confident (4.16 out of 5) that their campaign application used the correct sensors, and the system handled the use of multiple sensors well (4.05 out of 5). However, users felt feedback was an issue with data submissions (2.89 out of 5). Users commented that "I wasn't sure if my data was uploaded or not. I just assumed it worked." To address this issue, feedback mechanisms for crowdsensing and methods for effectively visualizing the sensor data on mobile phones are being explored.

In scenario B (Biketastic), users expressed significantly more confidence in MC Designer's usage of the sensors, feedback provided about the data being collected, and the handling of multiple sensors (questions 2, 4, and 5 of Table 1, respectively). Most significantly, user confidence that MC Designer was creating an application that served the purposed they expected (question 6 of Table 1) increased dramatically, implying that users became more comfortable with crowdsensing requirements and were able to create the application with ease after just one interaction with the tool.

Accuracy of the Created Applications. In our study, we define an "accurate" implementation as one in which the user incorporates all of the required sensors, and only those sensors, and inserting the appropriate text-based questions to address data that cannot be captured with sensors in scenario A (Meteor Count). In scenario A, only 2 of 19 users were able to create a mobile app that was exactly as the scenario described. The user's mistakes were most likely due to misinterpretations of the scenario and not an inability to configure the campaign. For example, multiple users entered the wrong number of text questions. They were capable of adding the questions in MCM, but they simply did not add everything the scenario described. A number of users also felt GPS was important to the scenario, despite no indication that GPS was required for the scenario. One user simply added the sensors he felt were "cool", almost entirely ignoring the scenario. Despite the low accuracy, users did feel as though they were creating the correct application, as evident by the 11 of the 19 users rating

Table 1. Perceptions of the campaign-creation process (1=strongly disagree, 5=strongly agree)

The mobile application generated by MC Designer:	Scenario A μ	Scenario A σ	Scenario B μ	Scenario B σ	p-value
1. used the sensors that I felt were needed for the scenario.	4.16	1.21	4.58	0.90	0.23
2. used the sensors appropriately for the scenario.	4.00	1.00	4.63	0.60	0.02
3. was able to collect data that was appropriate for the scenario.	3.95	1.13	4.42	0.96	0.17
4. provided me with feedback about my data.	2.89	1.20	3.58	1.17	0.08
5. handled the use of multiple sensors well.	4.05	0.91	4.53	0.84	0.10
6. served the purposes I expected.	3.58	1.30	4.53	0.70	0.01
Overall Average	3.77		4.38		0.04

"agree" or "strongly agree" to question 6 of Table 1, indicating MC Designer allowed them to create an application *they believed* was correct based on their understanding of the scenario.

In scenario B (Biketastic), 6 users were able to create the correct application, despite requiring twice as many sensors. Of the 19 users, 17 felt they created the correct application. Again, the large disparity between these findings are believed to be due to differing interpretations of the scenario and a lack of knowledge about the sensors, as opposed to an inability to add the sensor in MCM. For example, one user chose text-based entry as the appropriate method for capturing the roughness of the road. When asked why they chose that sensor, they responded, "I didn't know an accelerometer could capture the roughness of the road, so I didn't use it." These findings indicate MCM would benefit from hints that educate the campaign organizer about the features they are configuring, and points to the need for a wizard-like feature in MCM that will help the end-user programmer to validate their application in a user-friendly way.

Campaign Application Development Time. The time required to create the application with MC Designer was measured for each scenario (Figure 4). The average time to create an app was 6 minutes for Scenario A (Meteor Counter) and 4 minutes for scenario B (Biketastic). These results indicate that users are easily able to create a mobile application, even in their first experience with MC Designer. Figure 4 also compares user creation time to their pre-survey rating of their comfort level with creating software. Users who rated their comfort level at the lower end of the spectrum did not show a significant difference from users who rated higher, indicating that a lack of perceived technical competence did not impede the users' abilities to create an application using MC Designer. As one user noted, "I don't use apps so was coming at this as someone with no experience as a user and was still able to use the app maker."

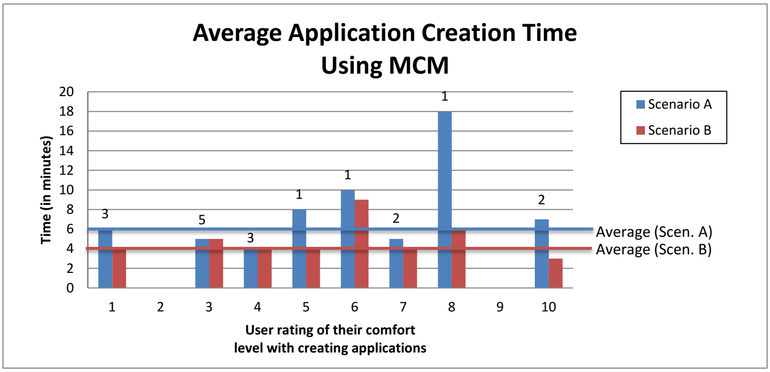


Fig. 4. Users' average creation time and comfort level with creating new software for scenario A (Meteor Counter) and scenario B (Biketastic). Bars are annotated with the number of users who rated themselves at a corresponding comfort level.

5.2 Part II: Comparing Custom Apps to MC Designer Apps

Part two of the user study is intended to evaluate expressiveness, captured by the user's comparison of the apps created using MC Designer and custom-built applications. Table 2 summarizes the results.

Comparison of Custom Apps and MC Designer Apps. In scenario A, both the Meteor Counter and MC Designer applications included an audio sensor and text-based input, but used them in slightly different ways. Meteor Count does not provide the user with a start or stop button to control recording audio. Recording begins when the user starts the data capture session, can be paused, and ends when the user uploads the contribution. The MC Designer app allows the user to explicitly start and stop the audio recording, but does not include the pause feature. Each press of the start button creates a new data collection session. For text-based input, Meteor Counter provides a slider option for entering predefined options for sky conditions and star visibility. To count meteors, the Meteor Counter app provides the user with a button to press each time a meteor is spotted. The MC Designer app captures these three entries by allowing the user to enter text.

Scenario B (Biketastic) included the accelerometer, audio, camcorder, camera, and GPS sensors. Again, differences exist between the way the Biketastic application and the MC Designer app incorporate each sensor. The Biketastic app collects accelerometer readings as a background task, and gives the user no feedback this data is being collected, whereas MC Designer displays the most current readings to the user. Similarly, audio is captured in the background in Biketastic, while MC Designer clearly displays "Recording audio" to the user. Camcorder and camera functions are very similar for the two apps; both use the built-in camera and camcorder APIs offered by the Android SDK. However, since riders should not be using these sensors while riding the bicycle, Biketastic automatically pauses the GPS, audio, and accelerometer sensors while camera

Table 2. Custom vs. MC Designer-generated apps (1=strongly disagree, 5=strongly agree)

	Custom	Clone	p
Scenario A - Meteor Counter			
1. Overall impression about using the application to collect data	4.42	3.89	0.05
2. Your impression about collecting data using the text-based input	3.95	4.05	0.49
3. Your impression about collecting data using the audio sensor	4.05	4.16	0.75
4. Which application would you prefer to use to collect data	50%	50%	
Scenario B - Biketastic			
5. Overall impression about using the application to collect data	4.00	4.05	0.84
6. Your impression about collecting data using the accelerometer	3.42	4.32	0.02
7. Your impression about collecting data using the audio sensor	4.05	4.21	0.60
8. Your impression about collecting data using the camera sensor	4.42	4.58	0.53
9. Your impression about collecting data using the camcorder sensor	4.32	4.42	0.72
10. Your impression about collecting data using the GPS sensor	3.84	4.37	0.16
11. Which application would you prefer to use to collect data	53%	47%	

functions are in use; MC Designer continues to collect this data. Also, Biketastic allows the user to upload multiple images or videos per a single data collection session; MC Designer allows only one per session. Lastly, GPS readings are converted to miles per hour and total distance when displayed to the user in Biketastic; MC Designer displays the most recent GPS location.

User Evaluations of Expressiveness. First, users were asked to compare the sensors used to collect data (i.e., "Rate your impression about the application's ability to collect data using the audio sensor") in each application. Overall, users rated the applications as equally able to capture data with two sensors (questions 2 and 3 in Table 2). These findings suggest that MC Designer can create apps that are *as good as* custom-built crowdsensing applications for capturing data.

Users were also asked to rate their overall impression of each application (question 1 in Table 2). For scenario A, the custom application rated significantly higher than the MC Designer app. Users were also asked to select the application they would prefer to use for data collection (question 4 of Table 2); the MC Designer app was selected by 50% of the users. When asked why one app was preferred, responses included "They are both good apps that capture data, you just have to play around with both to see how well they work," and "Basically the only difference is how it looks..." and "Both worked well but the NASA brand name made me biased." These results indicate that users were not only considering the application's ability to collect data when comparing the applications, but also the look-and-feel of the app. Clearly, crowdsensing applications that are built for a particular purpose will likely have a highly specialized user interface design that will tend to be more attractive. This is a trade-off that we expect, but we do plan to provide more options for customizing the UI with MC Designer in the future.

For scenario B, when asked to rate their overall impression (question 5 in Table 2), users had equally favorable impressions for each application. Question 11 aligns with these findings, with 47% of users stating that they preferred the MC Designer app for data collection. Responses from users included "The clone was more straightforward and I like the real time accelerometer data," and "The clone provided immediate feedback about the data the sensors were collecting, which is useful for an app like this," and "The clone seems to be just a simplified version of the actual app, which some people prefer."

The application's use of each sensor was also evaluated. Users rated use of the audio, camera, camcorder, and GPS sensors nearly equally across the MC Designer created application and the custom implementation in terms of their ability to collect data (questions 7-10 in Table 2). Users rated the accelerometer sensor (question 6 in Table 2) in the MC Designer app significantly higher than in the custom Biketastic app. User interviews indicate this is largely due to the fact that no real-time feedback about accelerometer data is provided by the Biketastic application. These findings indicate that for scenario B, MC Designer succeeded in being able to create an *equally functional* mobile application for data collection.

Lastly, users were surveyed on their overall experience while using MC Designer (Table 3). On average, users were pleased with MC Designer and felt they could create crowdsensing campaigns with ease. Users were very confident that MC Designer is able to create a mobile application for data collection (4.68 out of 5), and applications created using MC Designer use the phone sensors correctly (4.79 out of 5). Most importantly, users felt that MC Designer provided an easy-to-use interface for creating mobile applications (4.63 out of 5).

5.3 User Study Assumptions and Limitations

The user study was conducted under the following assumptions and limitations:

- This study focused on the user's ability to create a crowdsensing application that could be used to enable a sensor, capture a digital data sample, and upload the sample. Users were not asked to give input for recruitment and sensor sampling parameters.
- Users were provided with a description of each sensor in MC Designer (e.g., "the accelerometer measures motion in 3 dimensions, including shaking, turning, and rotating"). Users were then asked to apply sensors to the two scenarios. However, users often used a different sensor than expected (e.g., recording the roughness of the road with text-entry instead of the accelerometer). This likely contributed to the disparity between the accuracy and the user's perception of accuracy for the created application. We did not evaluate how well the users understood the scenario or sensor descriptions.
- Users did not submit data to the original apps. (We did not want users submitting synthetic/test data to the real Meteor Counter and Biketastic campaigns.) Future work will use a large-scale deployment of an MC Designer generated app to evaluate the utility of data collected.

Table 3. Perceptions of MC Designer (1=strongly disagree, 5=strongly agree)

Survey Question	Average Rating
1. I found the MCM application easy-to-use.	4.42
2. MCM is able to create a mobile application	4.68
3. MCM is able to incorporate phone sensors into a mobile application.	4.79
4. The mobile applications generated by MCM are the same as what I was expecting.	4.16
5. MCM allows me to define everything about the scenarios that I felt was important.	3.89
6. MCM provided me with an easy-to-use interface for creating a mobile application.	4.63
7. MCM included all the sensors I wanted to use for the scenarios.	4.53
8. MCM let me input all the information about the mobile application that I wanted to enter.	4.32

6 Conclusions and Future Work

We have presented MC Designer, a tool that enables end-users to create and manage crowdsensing campaigns from a mobile phone. MC Designer has a core set of constructs for building crowdsensing apps, and is designed with an extensible API with hooks for implementing emerging solutions in active areas of research, such as privacy [35,16], data analysis tools [33], and sensor and data fusion [3,36]. Since MC Designer is open source, researchers can integrate their own implementations of these concepts into the tool. Our evaluation of MC Designer indicates that users with varying levels of programming skill are able to use MC Designer to create an application that meets their expectations in an average of five minutes. Furthermore, the applications created by MC Designer were perceived as equivalent to or better than custom-built implementations of existing crowdsensing applications.

References

1. Aanensen, D., Huntley, D., Feil, E., Spratt, B., et al.: Epicollect: linking smartphones to web applications for epidemiology, ecology and community data collection. PLoS One 4(9), 6968 (2009)
2. Amazon.com, Inc. Amazon Mechanical Turk (June 2011), `https://www.mturk.com/mturk/welcome`
3. Beach, A., Gartrell, M., Xing, X., Han, R., Lv, Q., Mishra, S., Seada, K.: Fusing mobile, sensor, and social data to fully enable context-aware computing. In: Proc. of HotMobile (2010)
4. Center for Embedded Networked Sensing. What's Invasive! community data collection (2011), `http://whatsinvasive.com/`
5. Das, T., Mohan, P., Padmanabhan, V.N., Ramjee, R., Sharma, A.: PRISM: platform for remote sensing using smartphones. In: Proc. of MobiSys, pp. 63–76 (2010)

6. Eisenman, S.B., Miluzzo, E., Lane, N.D., Peterson, R.A., Ahn, G.-S., Campbell, A.T.: BikeNet: A mobile sensing system for cyclist experience mapping. ACM Transactions on Sensor Networks 6(1), 1–39 (2009)
7. Ganti, R.K., Pham, N., Ahmadi, H., Nangia, S., Abdelzaher, T.F.: GreenGPS: a participatory sensing fuel-e_cient maps application. In: Proc. of MobiSys, pp. 151–164 (2010)
8. Gaonkar, S., Li, J., Choudhury, R.R., Cox, L., Schmidt, A.: Micro-Blog: sharing and querying content through mobile phones and social participation. In: Proc. of MobiSys, pp. 174–186 (2008)
9. Griswold, W., Shanahan, P., Brown, S., Boyer, R., Ratto, M., Shapiro, R., Truong, T.: ActiveCampus: experiments in community-oriented ubiquitous computing. Computer 7(10), 73–81 (2004)
10. Hartung, C., Lerer, A., Anokwa, Y., Tseng, C., Brunette, W., Borriello, G.: Open data kit: Tools to build information services for developing regions. In: Proc. of ICTD, p. 18 (2010)
11. Heggen, S., Omokaro, O., Payton, J.: MAD Science: Increasing engagement in STEM education through participatory sensing. In: Proc. of UBICOMM (October 2012)
12. Hicks, J., Ramanathan, N., Kim, D., Monibi, M., Selsky, J., Hansen, M., Estrin, D.: AndWellness: an open mobile system for activity and experience sampling. In: Proc. of Wireless Health, pp. 34–43 (2010)
13. Huang, K.L., Kanhere, S.S., Hu, W.: Are you contributing trustworthy data?: The case for a reputation system in participatory sensing. In: Proc. of MSWiM (2010)
14. Joki, A., Burke, J., Estrin, D.: Campaignr: A framework for participatory data collection on mobile phones. Technical report, UCLA Center for Embedded Network Sensing (2007)
15. Kanjo, E., Bacon, J., Roberts, D., Landshoff, P.: MobSens: Making smart phones smarter. Pervasive Computing 8(4), 50–57 (2009)
16. Kapadia, A., Kotz, D., Triandopoulos, N.: Opportunistic sensing: Security challenges for the new paradigm. In: Proc. of COMSNETS, pp. 1–10 (January 2009)
17. Kim, S., Mankoff, J., Paulos, E.: Sensr: evaluating a exible framework for authoring mobile data-collection tools for citizen science. In: Proc. of CSCW, pp. 1453–1462 (2013)
18. Kim, S., Robson, C., Zimmerman, T., Pierce, J., Haber, E.M.: Creek watch: Pairing usefulness and usability for successful citizen science. In: Proc. of CHI, pp. 2125–2134 (2011)
19. LeBaron, G.: The 112th Christmas Bird Count. American Birds 66, 2–8 (2012)
20. Mediated Spaces, Inc. The WildLab: Use mobile technology to explore, discover, and share the natural world, `http://thewildlab.org`
21. Mendez, D., Perez, A., Labrador, M., Marron, J.: P-sense: A participatory sensing system for air pollution monitoring and control. In: Proc. of PerCom Workshops, pp. 344–347 (2011)
22. Miluzzo, E., Lane, N., Eisenman, S., Campbell, A.: CenceMe: Injecting sensing presence into social networking applications. In: Proc. of Euro SSC, pp. 1–28 (October 2007)
23. MIT Media Lab. App inventor for android, `http://www.appinventor.mit.edu/`
24. Mohan, P., Padmanabhan, V.N., Ramjee, R.: Nericell: using mobile smartphones for rich monitoring of road and traffic conditions. In: Proc. of SenSys, pp. 357–358 (2008)

25. Mun, M., Reddy, S., Shilton, K., Yau, N., Burke, J., Estrin, D., Hansen, M., Howard, E., West, R., Boda, P.: PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In: Proc. of MobiSys, pp. 55–68 (2009)
26. NASA. Meteor Counter (2011), http://meteorcounter.com/
27. National Audubon Society. The Christmas Bird Count (2012), http://birds.audubon.org/christmas-bird-count
28. NEON and the Chicago Botanic Garden. Project Budburst (2011), http://neoninc.org/budburst/
29. Networked Organisms. Project Noah, http://www.projectnoah.org
30. Ra, M., Liu, B., La Porta, T., Govindan, R.: Medusa: A programming framework for crowd-sensing applications. In: Proc. of MobiSys, pp. 337–350 (2012)
31. Ravindranath, L., Thiagarajan, A., Balakrishnan, H., Madden, S.: Code in the air: simplifying sensing and coordination tasks on smartphones. In: Proc. of Hot-Mobile (2012)
32. Reddy, S., Estrin, D., Srivastava, M.: Recruitment framework for participatory sensing data collections. In: Floréen, P., Krüger, A., Spasojevic, M. (eds.) Pervasive 2010. LNCS, vol. 6030, pp. 138–155. Springer, Heidelberg (2010)
33. Reddy, S., Parker, A., Hyman, J., Burke, J., Estrin, D., Hansen, M.: Image browsing, processing, and clustering for participatory sensing: lessons from a DietSense prototype. In: Proc. of EmNets, pp. 138–155 (2010)
34. Reddy, S., Shilton, K., Denisov, G., Cenizal, C., Estrin, D., Srivastava, M.: Biketastic: sensing and mapping for better biking. In: Proc. of CHI, pp. 1817–1820 (2010)
35. Shilton, K.: Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection. Communications of the ACM 52(11), 48–53 (2009)
36. Spanos, D.P., Murray, R.M.: Distributed sensor fusion using dynamic consensus. In: World Congress of the International Federation of Automatic Control (2005)
37. Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., Eriksson, J.: VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In: Proc. of SenSys, pp. 85–98 (2009)
38. Tillmann, N., Moskal, M., de Halleux, J., Fahndrich, M.: TouchDevelop: programming cloud-connected mobile devices via touchscreen. In: Proc. of ONWARD, pp. 49–60 (2011)