World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:16, No:3, 2022

The CommonSense Platform for Conducting Multiple Participant Field-Experiments Using Mobile-Phones

Y. Hoffner, Y. Rusho, S. Rubach, S. Abargil

***Abstract*—**This paper presents CommonSense, a platform that provides researchers with the infrastructure and tools that enable the efficient and smooth creation, execution and processing of multiple participant experiments taking place outside the laboratory environment. The platform provides the infrastructure and tools to accompany the researchers throughout the life cycle of an experiment – from its inception, through its execution, to its processing and termination. The approach of our platform is based on providing a comprehensive solution, which puts emphasis on the support for the entire life-cycle of an experiment, starting from its definition, the setting up and the configuration of the platform, through the management of the experiment itself and its post processing. Some of the components that support those processes are constructed and configured automatically from the experiment definition.

***Keywords*—**Mobile applications, mobile experiments, web experiments, software system architecture.

I. Introduction

THE aim of our project was to provide support for conducting social and psychological experiments on a large scale in their natural environment. Our CommonSense platform creates the environment in which researchers can create, execute and process the results of experiments that take place outside the laboratory space, engaging multiple participants over extended periods of time. The platform offers a number of novel features. In addition to supporting researchers through the entire life cycle of an experiment, the tools together with the experiment definition facilitate the following:

- Automatic generation of certain components of the system such as the format of the mobile phone messages and the database schema for storing them for later analysis.
- Automatic configuration of the data collection points and the databases that can be derived from the experiment definition data such as the number of participants, length of the experiment, number of sensors used and rate of sampling them.

In addition, it is possible to modify, during the lifetime of an ongoing experiment, the instructions which define what information should be acquired and when from the participants and their surroundings.

Hoffner Yigal is with Shenkar College of Engineering, Design and Art. Anne Frank St 12, Ramat Gan, Israel (corresponding author, e-mail: yigal.hoffner@shenkar.ac.il).

Rusho Yonit, Rubach Shay, and Abargil Sapir are with Shenkar College of Engineering, Design, and Art. Anne Frank St 12, Ramat Gan, Israel.

The structure of the paper is as follows. Section II is a survey of related work. Section III describes the three main processes supported by the platform, the roles associated with them and a description of an experiment life-cycle. Section IV describes the platform sub-systems and their interfaces. Sections V-VII describe the Experiment Preparation, the Experiment Execution and the Experiment Processing systems respectively. Section VIII describes how the management of the experiment can be done while it is being carried out. Section IX covers various aspects of the design and implementation of the platform. Section X provides notes on the evaluation of the platforms. Finally, Section XI provides a summary of the project and some of the conclusions drawn from it.

II. Related Work

The idea of using mobile phones for collecting behavioral data on a large sale, as part of the daily life of multiple participants has been well documented [1]-[3].

Several projects have set out to exploit the potential of mobile phones built-in sensors for conducting long-term multiple-participant field experiments of a psychological and social nature.

BeTelGeuse is a platform for gathering and processing situational data [4]. The platform supports collecting data from various sources and the use of an inference engine in order to derive higher level context from sensor data.

EmotionSense is a mobile-phones based platform - a fully context-aware programmable mobile sensing system for social psychology research [5]. It supports co-location detection and emotion and speaker recognition.

CenceMe [6], [7] detects personal sensing presence such as walking, being in conversation or exercising in the gym from classifiers which are distributed between the phones and the backend servers in order to achieve scalable inference.

MetroSense was an early project in the area of people-centric sensing at scale [8]. Its focus was on the mobility aspect of such experiments, the heterogeneity of the involved infrastructures and the problem of maintaining communication over large areas.

III. The CommonSense Platform Overview

This section provides an overview of the structure of the platform, introducing its sub-systems, the roles that are related to those systems and the tools that role holders need in order

to carry out their tasks. Lastly, the life cycle of an experiment is outlined and explained.

A. The Overall CommonSense Platform Structure

The platform is divided into two major parts - the researchers' side and the participants' side (Fig. 1).

1) Researchers' side - The Experiment Organization system: This includes all the processes that are associated with preparing and downloading the experiment to the mobile phone system, and the post-experiment processing. This part oversees the different stages of the experiment from the researchers' side. The organization side can be further divided into:
- Experiment Preparation system: This includes the creation of the experiment and its definition.
- Experiment Management system: This includes the real-time control of the experiment in case the experiment is not progressing as expected.
- Experiment Processing system: This includes the collection and storage of data and its later processing.

2) Participants' side - The Experiment Execution system: This includes the mobile-phone with the application software downloaded from the experiment preparation system. Its task is to carry out the measurements of the participants' activities, their interactions with other participants and the state of the environment.

B. The Experiment Platform Roles and Their Tools

With the exception of participant recruitment, the system provides tools to support the functions that each role requires in order to carry out their tasks. The roles associated with the experiment life cycle are divided into three main categories:

1) Experiment Researchers - this category can be further divided into:

a) Experiment Definer: Responsible for the preparation steps of the experiment. Tools are necessary for the following activities:
- Define the general aspects of the experiments.
- Write configuration and interpreter scripts for the mobile phone.
- Download the script to the mobile phone and test the system.
- Write data collection configuration scripts.

b) Experiment Analyzer: Responsible for extracting patterns from the data and making sense of it. This means that tools for displaying data, filtering, sorting, aggregating and cross-correlating data are needed.

2) Experiment Manager: Responsible for modifying the experiment script if the data coming from the experiment indicates that there is a need to switch on additional sensors or to increase/decrease the rate of sending data.

3) Experiment Participant: Responsible for carrying out the experiment according to the agreement with the researchers and the instructions sent to it from the experiment definer. There is no need for extra tools; the mobile phone must have the necessary sensors and be able to communicate with the user when specific responses are needed as input.

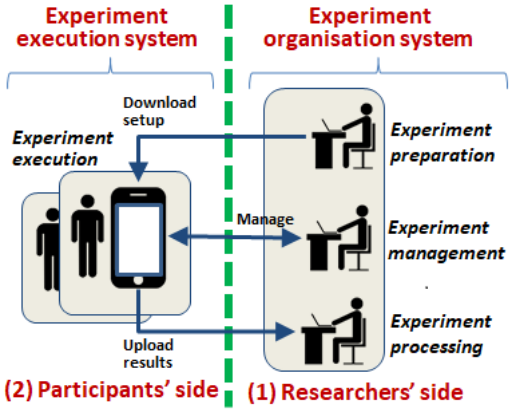

Fig. 1 The general layout of the CommonSense platform showing its main components – the experiment organization and the experiment execution systems

C. The Experiment Platform Life-Cycle

The experiment life cycle (Fig. 2) consists of the series of stages, which an experiment goes through from creation to termination.

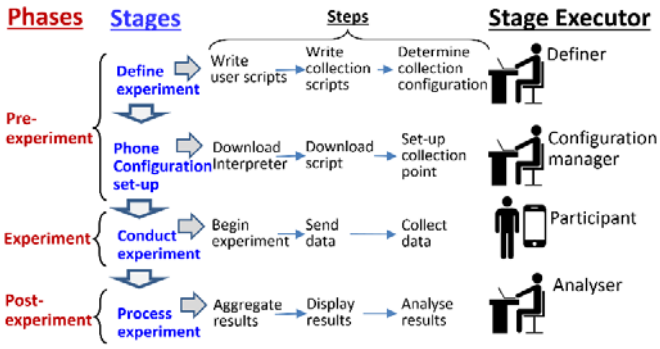

Fig. 2 The experiment platform life cycle showing the different phases and stages between the creation and termination of an experiment

There are three major phases in the life cycle of an experiment:

i. Pre-experiment phase: All the preparation of the various aspects of the experiment.
ii. Experiment phase: The participants actually conducting the experiment by participating in it.
iii. Post-experiment phase: The analysis of the collected data from the participants.

IV. The Platform Sub-Systems and Their Interfaces

The main interfaces between the major components are best explained in terms of the information that flows between them (Fig. 3):

1) The Experiment Preparation system to the Experiment Execution system (the mobile phone application). The main component passed through this interface is the script that contains the instructions for getting data from the

sensors.

2) The experiment management system to the Experiment Execution system (the mobile phone application). This involves observing the progress of the experiment and sending it a new script if necessary.
3) The Experiment Execution system to the experiment processing system collection point. This takes the form of messages sent from the mobile phone to be stored for later processing.

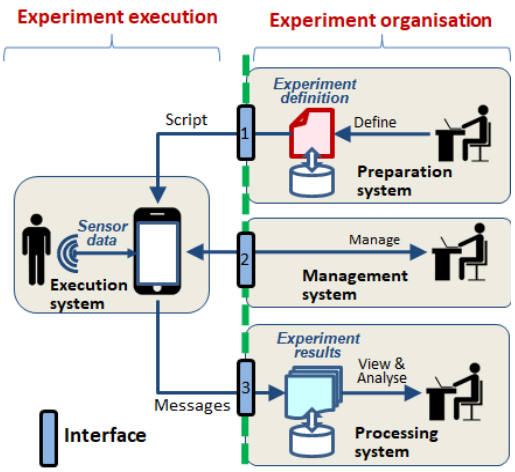

Fig. 3 The Interfaces between the systems that make up the platform

V. The Experiment Preparation System

This part of the platform deals with the overall definition of the experiment in terms of scope, goals, the target group, etc. In addition, the experiment definition has to specify the following items precisely (this is not an exhaustive list):

1) Types of events of interest, which in turn indicate:
- What type of data is to be collected - which defines the sensors to be used and the associated data types. This in turn defines the sensor configuration.
- When is it to be collected - defines the rules which specify the conditions under which:

a) Sensor readings are taken.
b) Readings are aggregated and recognized as more complex events.
c) Sensor readings or event data are sent on to the experiment processing system.

2) The destination of the data from the mobile phone: It encompasses the collection-point to which the messages are sent before they are stored in a database.
3) The format of the messages sent from the mobile phone to the collection point: This depends on the type of data being collected.
4) The number of participants: This determines the load on the data collection point and may indicate that the collection points have to be distributed manner.
5) The start and length of the experiment: For managing the mobile phone application.

A. Experiment Preparation Tools and Issues

The platform offers the Experiment Editor Tool, which is responsible for creating a new experiment. This component is the entry and starting point of the platform life-cycle. Among other items, the Editor lets the researcher to write the Rule Script by specifying which data type should be collected by the participants and under which conditions.

The combination of data type to be collected and conditions to be fulfilled in order to do so will form a 'Rule', or 'Experiment Rule'. A set of one or more rules define an experiment Script and will later be 'Interpreted' by the Interpreter. In effect, the rules of the Script, together with the behavior of the participant, set the flow of the experiment.

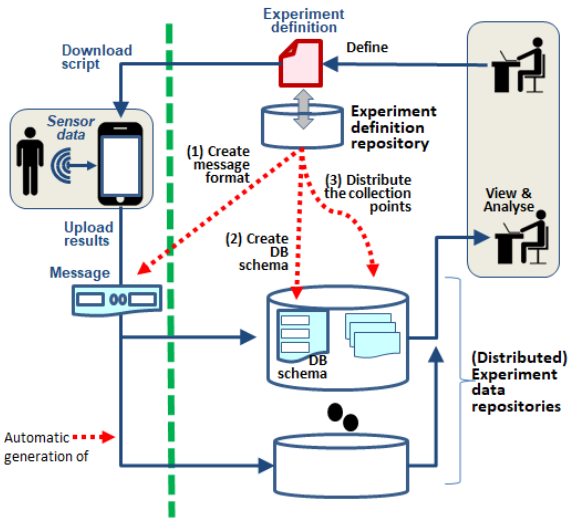

Fig. 4 Using the definition of the experiment to generate the format of the mobile-phone messages, the DB schema and to determine the distribution of the collection points

B. Automatic Generation of Components from the Experiment Definition

The experiment definition together with the script can be used to automatically generate several items as shown in Fig. 4:

1) The format of the messages from the mobile phone to the collection point in the experiment processing system.
2) The DB schema for storing the data from the incoming messages (carried using the above defined format).
3) The configuration of the data collection points where the data from the mobile phone are collected.

The data collection point configuration depends on whether load balancing problems require the distribution of the data collection points over more than one server. This can be determined by estimating the rate of sending data and the quantity of data that will be accumulated during an experiment. This can be derived from the experiment definition data:

*Experiment data quantity = number of participants ∗ number of sensors ∗ experiment duration ∗ rate of sampling * size of data types*

Experiment data rate = number of participants ∗ number of sensors ∗ rate of sampling/sec

It is highly unlikely that the users of the platform, being primarily psychologists and sociologists, will be technically savvy to the point of being able to deal with all the fine technical details of the experiment life-cycle. The ability to use the parameters of the experiment definition to generate components that are used throughout the experiment is therefore of considerable help.

VI. The Experiment Execution System: The Mobile Phone Application

This part of the platform, which runs on a mobile phone, has the task of executing an experiment as dictated by the script, downloaded earlier from the Experiment Preparation system. A Script Interpreter runs the experiment according to the rules in the script, by obtaining data from the sensors and ultimately sending it to the collection point of the experiment processing system as shown in Fig. 5.

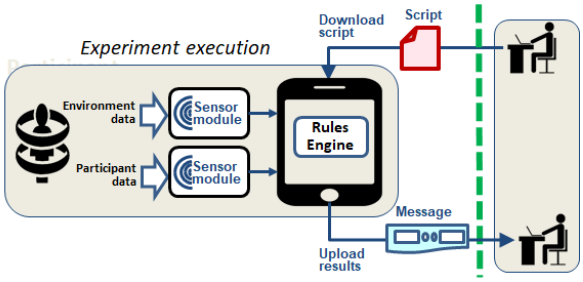

Fig. 5 The data sent to the mobile phone includes a script that tells the Interpreter when to obtain sensor data and send it in a message to the collection point

A. The Rules Engine: The Interpreter, the Script and the Experiment State

The Rules Engine system (Fig. 6) is responsible for the management of the experiment. It consists of three main components:

1) The Script Interpreter is the main component that is responsible for acquiring the relevant data from the sensors, and ultimately sending it to the execution processing system. The Interpreter is rule-based, so that it executes rules presented to it from the Script, downloaded earlier from the Experiment Preparation system. The Interpreter also saves some data locally - in a container called the Experiment State, so that the data can be used as context for interpreting the rest of the experiment.
2) The Experiment State can be used to hold different types of information:
a) Personal state data: Data of physiological nature, description of feelings, keywords from a conversation, etc.
b) Socially oriented state: Proximity of people or friends, social interaction, etc.
c) Environment related state: Locations, external conditions such as weather, etc.
3) The Script consists of rules, which are built from condition and action pairs, where the action is carried out if and when the condition is true. The Interpreter fetches the instructions from a script downloaded to the Interpreter prior to starting an experiment. The Interpreter consults the state of the experiment together with data from sensors to decide what actions should be taken. An action can be another reading from a sensor, saving data in the Experiment State or sending a message to the experiment data collection point.

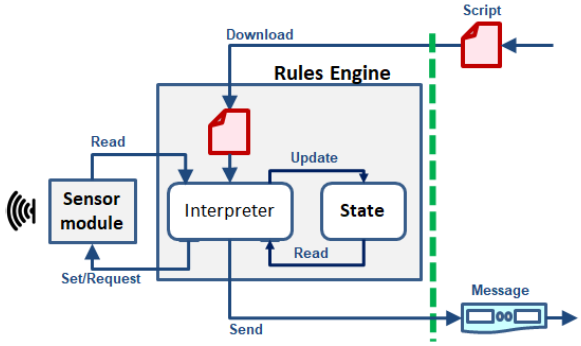

Fig. 6 The Rules Engine consists of the Interpreter that activates the sensors as specified by the rules in the Script. The Interpreter also updates and uses the experiment state to make its decisions

B. The Script language

The definition of the experiment is based on the script – which is executed by the Script Interpreter. The script language contains Rules; each rule consists of a Conditions and a related Action. The general form of a rule is:

Condition -> Action // When the **condition** becomes true, // the **action** is triggered.

An example of a rules is shown in Fig. 7.

A **Condition expression** can consist of a combination of:

1) A Condition can have a comparison between values where the values can be:
(a) A reading from a sensor or a value from past readings (the Experiment State).
(b) Clock reading - Date/time in order to facilitate activations at prescribed times and intervals.
2) Conditions connected by logical connectives such as conjunction and disjunction.

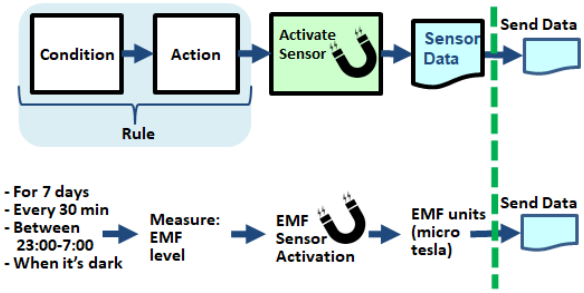

Fig. 7 An example of a Rule consisting of a Condition and an Action

An **Action** can be any or a combination of:

a. Requesting a reading from a sensor - either directly or by asking the participant to provide input;
b. Combining several readings into an aggregated event.

c. Sending data as a message to the collection point.

An example of the kind of rule the script can have is the following:

if (*DayLight* == **true**) **AND** (*CurrentTime* > 15:00PM)
→ take a GPS snapshot // "read" from the GPS Service.

C. The Interpreter and the Sensors – to 'Push' or to 'Pull'

There are several ways to implement the interaction between the Interpreter and the sensors:

1) PUSH approach: The Sensors can initiate the sending of the data to the interpreter.
2) PULL approach: The Interpreter can request a reading from the sensor.
3) PUSH/PUL hybrid approach: Both sides may initiate a sampling and the data transfer can be done by
a) Shared memory: The sensor puts the data in a shared memory location - where the Interpreter can pick it up.
b) Mutual invocations: When the Interpreter requires a value – it requests data from the sensor. When the sensor wants to send data to the Interpreter – it invokes the Interpreter.

Our system uses the Pull approach, where the Interpreter determines when to get a reading from a sensor and initiates it.

D. The Sensors and Sensor Modules

There is a growing set of sensors which are becoming an integral part of a smart phone or that can be added relatively easily. The sensors can be used to obtain different types of information:

1) Participant related data: Data of physiological nature, recording of conversations, description of feelings, etc.
2) Socially oriented data: Proximity of people or friends, social interaction, etc.
3) Environment related data: Locations, external conditions such as weather, etc.

The only down-side of this plethora of sensors, is the fact that they may vary in different phones, imposing phone specific modification of the code accessing the devices.

VII. The Experiment Processing System

This part of the platform has several functions (Fig. 8):

1) Receive the messages from the mobile-phone, extract the data from them and store it in a database for later retrieval.
2) Display the arriving messages to the viewer, either at a later stage or in real time as the messages arrive.
3) Process the data by employing filtering, aggregating, sorting, counting and translating the data into a representation that is more meaningful for analyzing the results of the experiment.

The *Experiment Data viewer and analyzer dashboard* resides in a Web Application and is responsible for presenting the experiment information, such as its current participants and status accompanied with all of the collected data. The dashboard also enables researchers to view statistics and visualizations of the acquired data.

Collected data can be viewed via the researcher dashboard and even downloaded as *xls* format for external supported analysis tools.

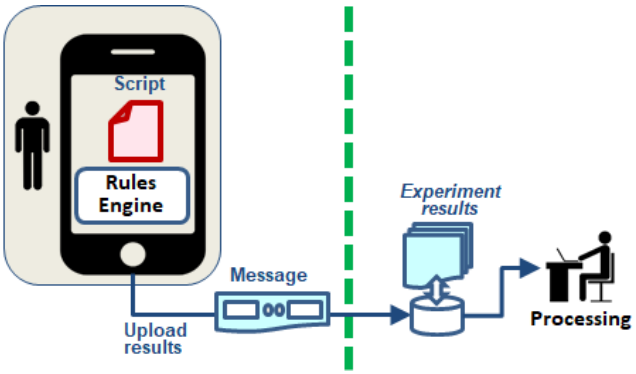

Fig. 8 The processing system retrieves the data from the database and presents it for analysis

VIII. Dynamic Runtime Experiment Management

Conditional statements already incorporate a certain level of management in being able to decide what will be done, based on the state of the experiment. However, there are limits to how flexible this can be without making the scripts too long and overly complicated. There are also situations where the experiment owners may conclude that the experiment is not going according to plan and therefore requires modifications.

A possible way to incorporate dynamic management of the experiment in an explicit form is shown in Fig. 9. This entails a channel for obtaining data from the experiment and a channel for controlling the experiment. Management carried out during an ongoing experiment is based on using information obtained from the mobile-phone, to decide whether anything in the experiment requires modifying, and sending the appropriate instructions or a new script to the Script Interpreter in the mobile-phone. Management therefore requires the following:

1) The ability to observe those aspects of the experiment which are relevant for determining how the experiment is progressing.
2) A control interface, where instructions can be administered in order to modify the managed entities.
3) Safe points in the experiment flow at which the changes can be applied without adversely interfering with the progress of the experiment.

While this appears straightforward, it is the scope of the changes that determines how difficult this is in practice. Changes made to Sensor configuration or the interval at which the sampling takes place are considered trivial. A change to a specific rule or to a number of rules may be complicated by the fact that it is necessary to find a quiescent experiment point at which the change can be injected without adverse side effects, such as disrupting the flow of the experiment or rendering some of the collected data useless.

There are a number of issues that have to be considered and if incorporated into the design of the system can make the process significantly easier.

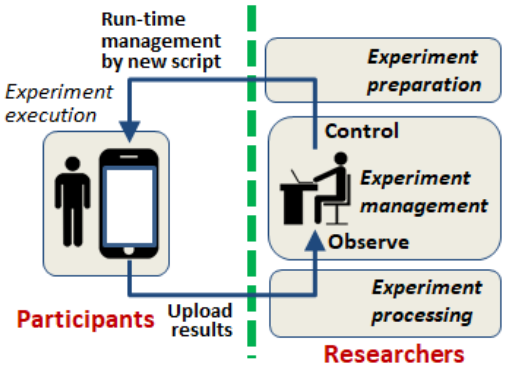

Fig. 9 Using data from the mobile phone as feedback, to decide how to control the application

One possible way to implement the management is shown in Fig. 10:

1) Use the existing data channel from the mobile-phone: If the data flowing from the experiment contain the information required for management, it can be channeled to the experiment manager to enable the necessary decision making. If this information is not part of the message definition – it can be piggybacked on the messages. This has the advantages that it exploits the existing data channel from the experiment, adding no overheads to the design of the platform except adding message formats for carrying management data. The only requirement it imposes is that the data being collected anyway have to include what is needed to make management decisions.
2) Use the existing script download channel: Change the script using the same channel that is used to download the script to the mobile phone. There is however a need to modify the Interpreter to be able to accommodate the changes.

IX. CommonSense Design and Implementation

This section provides a description of the implementation of the key aspects of the platform, the development environment and programming languages used.

A. Client and Server-Side Implementation

The back-end server side – the Experiment Organisation system is implemented using the Heroku cloud solution. It is written in Node.js and data are saved in the NoSQL MongoDB Database.

Two types of clients are involved with the platform: The Researchers and the Participants.

The Researchers' client-application - where the researcher defines and manages the experiment, is a web application implemented with JavaScript (JS) and an Angular library. Angular is particularly suitable for this application as it supports a data binding technique that enables synchronization between the properties of two distributed objects and thus facilitates the transfer of data in real time.

The Participants' client-side mobile phone uses Java and Kotlin for the mobile application support. The mobile client-side and the server back-end communicate with each other using a dedicated REST API. Images and other multimedia files are stored on Cloudinary.

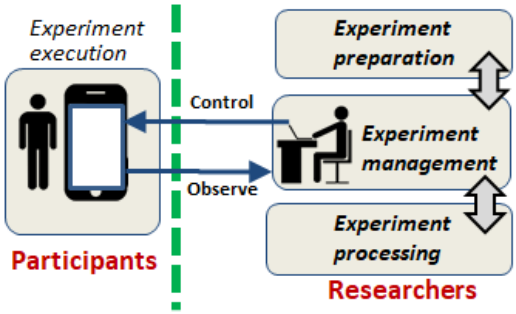

Fig. 10 Exploiting the available channels for data collection and control of the Interpreter in order to achieve the management of the experiment

The Service API of the Android OS provides a way to initiate a service - to run an application component that can perform long-running operations in the background. It is used by the Interpreter to create the required asynchronous tasks which will be invoked through and managed by the OS. This enables the Interpreter to activate and manage the available sensors in an asynchronous manner. The use of the shared service instead of spawning threads by the Interpreter, reduces overheads and also enables running more than one experiment in parallel.

The steps taken by the system are:

1) In the Experiment Preparation system, on the server side - a JSON object is created from the Script (obtained from the Experiment Definer) to be used by the mobile phone Interpreter.
2) In the Experiment Execution system, on the mobile application side - when the participant joins the experiment, the JSON object is fetched from the server with a GET request. The application parses its fields and uses the data to create an Experiment object for the Interpreter. The Interpreter is a Class which resides inside each Experiment Class and at this point, the Interpreter has all the data it needs to start the interpretation loop. This loop checks the condition part of each rule; when the condition is evaluated to true, the Interpreter will invoke the appropriate action as an asynchronous-tasks, which will subsequently activate a sensor and collect the data (through the Service API) from it.
3) The data are sent from the client side to the server using a POST request which is available in the REST API.

X. The CommonSense System Evaluation

In addition to carrying out tests with participants, we used the Postman's built-in feature, the Collection Runner, to simulate heavy traffic from multiple participants' phones to the collection point in the server. The Collection Runner performed POST requests at different rates and intervals: between 100 to 1000 POST requests were sent at intervals

ranging from 0.1 to 0.01 seconds. The response in each case was more than adequate.

In extreme cases where the heavy traffic from a large number of participants exceeds the capacity of the system, a Load Balancer can be applied to assist with balancing the concurrent traffic coming from multiple participants. This approach was tested and has shown that multiple mobile clients are able to efficiently acquire data using its sensors, while keeping a constant, stable connection with the server. The data are perfectly synchronized and the results are displayed instantly on a web application – the Dashboard. Moreover, due to a responsive approach, researchers can manage their experiments and view its progress from their desktop or mobile device via browser.

A. The CommonSense Platform Limitations

Mobile device sensors vary among phone manufacturers; therefore, one participant may not be suitable for participating in an experiment due to a non-supported or non-existing sensor which is mandatory in the experiment.

The current versions of the client-side mobile phone software are not portable. The implementation is a native android application, tested with Xioami MiA2 and Google Nexus 5. Changes will have to be made in order to be able to run the platform with iOS-based devices.

Most of the systems surveyed earlier have carried out some inferencing in order to identify high-level activities from the collected data. We have decided not to include inferencing capability in the Rules Engine setup of the initial platform implementation. However, there is no reason why such a facility cannot be added in future implementations.

XI. Discussion and Conclusions

In this paper we presented our platform which aims to support multiple participant field-experiments using mobile-phones. The platform provides the infrastructure and tools to aid the researchers throughout the life cycle of an experiment. The different parts which support the various stages of an experiment are designed and implemented so that in addition to working seamlessly, the system is capable of using the experiment definition in order to automatically generate certain components of the system. While this feature helps with experiment development, it does not address a major problem of experiment definition – the writing of the Script. This remains the Achilles heel of such systems as it requires extensive programming ability.

References

[1] G. Miller, "The Smartphone Psychology Manifesto," Perspect. Psychol. Sci., vol. 7, no. 3, pp. 221–237, May 2012.

[2] K. Hochberg, J. Kuhn, and A. Müller, "Using Smartphones as Experimental Tools—Effects on Interest, Curiosity, and Learning in Physics Education," J. Sci. Educ. Technol., vol. 27, no. 5, pp. 385–403, Oct. 2018.

[3] J. Kuhn and P. Vogt, "Applications and Examples of Experiments with Mobile Phones and Smartphones in Physics Lessons," Front. Sensors, vol. 1, no. 4, 2013.

[4] J. Kukkonen, E. Lagerspetz, P. Nurmi, and M. Andersson, "BeTelGeuse: A platform for gathering and processing situational data," IEEE Pervasive Comput., vol. 8, no. 2, pp. 49–56, 2009.

[5] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas, "EmotionSense: A mobile phones based adaptive platform for experimental social psychology research," in UbiComp'10 - Proceedings of the 2010 ACM Conference on Ubiquitous Computing, pp. 281–290. 2010.

[6] A. Campbell and S. Eisenman, "Cenceme: Injecting sensing presence into social network applications using mobile phones (demo abstract)," Demo at ACM MobiHoc 2008.

[7] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "CenceMe-Injecting Sensing Presence into Social Networking Applications," Smart Sensing and Context, Second European Conference, EuroSSC, 2007.

[8] S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "MetroSense Project: People-Centric Sensing at Scale," Work. World-Sensor-Web, pp. 6–11, 2006.