Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 98 (2016) 169 – 173

Procedia Computer Science

The 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016)

Ubiquitous Citizen Programming

Dave Mason

Ryerson University, Toronto M5B 2K3, Canada

Abstract

Modern society is increasingly mediated by computers. The quantity and diversity of data generated daily is growing at an astounding rate. While the number of people with programming ability is also growing, the percentage of the population with such capability remains in the single digits. This means that we are witnessing a growing gap between need and capacity. Some of the need can be met by packages and apps that provide canned or predefined analysis, but these restrict analyses to those anticipated by the creators of those packages and apps.

It is essential that in a data society nearly all citizens have a capacity to program. Therefore, it is critical that we find "programatic" analysis capabilty that is available and accessibly by most citizens.

Peer-review under responsibility of the Program Chairs

Keywords: Citizen Programming, Ubiquitous Computing, Data Society

1. Introduction

You're driving home one night and you hit a pothole. Glancing in the rearview mirror you notice that the streetlight is burnt out there. Then you notice another light out and hit another pothole. You wonder if there's some kind of correlation.

Today: you might contact a city newspaper and try to convince them it was an interesting story, so that they could hire a programmer to access the data and produce some kind of report – but more likely, you flake out in front of the TV and forget about it.

If you had an environment that allowed you to access a variety of data resources and easily manipulate that data, things might be different. You poke around on the WWW and find that the power company has a web page with a list of all the burnt-out lights, so you mark that. Then you find a poblic database from the city that contains the work-orders to repair broken pavement and you mark that. You write a little program to correlate the data from the power company and the city and display the information in a map. Sure enough you see a correlation, but you also notice that the worst situation appears to be in the poorer parts of town, so you pull in a spreadsheet from the census

E-mail address: dmason@ryerson.ca

Peer-review under responsibility of the Program Chairs

doi:10.1016/j.procs.2016.09.027

that contains income-by-electoral-disctrict data and sure enough, that's correlated too. In 15 or 20 mminutes you have transformed some raw, public data into real information that you can share with a news outlet or ask a politician to do something to fix!

While there are a few professional programmers who could do this analysis today, it would take them significantly more time than outlined above - so they would be much less likely to do it – and more importantly they might not have noticed the puzzle that motivated you to explore the data in the first place.

We believe that as the data available to us becomes more available, and the problems facing us become more challenging, it is critical that most citizens be able to process that data so that the serendipity of observations such as those in our scenario can be capitalized upon. We also believe that citizen oversight of governments and corporations in our increasingly complex and inter-connected world requires a broad-based capability to perform computation.

2. Citizen Programming

When we refer to Citizen programmers, we mean primarily amateur programmers that are programming not merely for education or entertainment, but are solving problems of their own devising, or analysing data to make decisions. In 2005 it was estimated [1] that in 2012 there would be 90 million end user programmers, including 13 million describing themselves as programmers. With the U.S. Bureau of Labor Stitistics describing 3 million people with the job description of programmer or feveloper, this indicates between 3 and 13 million people using traditional, text-based programming languages.

2.1. History

Over the years there have been many languages and environments that have facilitated a degree of programming for the amateur programmer. These environments have often been met with derision from professional programmers even as they have provided validation, entertainment, and capability to the amateur programmer.

2.1.1. Early Programming

The earliest programmers were, of course, not trained programmers. Rather they were mathematicians, engineers, or scientists who wanted to use the fast – if primitive – computers to solve their primarily mathematical problems. They were programming in machine code or assembly language – later FORTRAN – but the computers and hence the programs that ran on them were very small.

In 1959, COBOL was created as a first attempt to create a portable language that would be accessible to a large cohort of business analysts with limited formal training in programming. It was relatively successful in this goal with over 1 million people learning to program in it.

In 1964, BASIC became available at Dartmough College, and versions of it became available on virtually every personal computer produced in the succeeding decades. It became the gateway language for thousands of future programmers [2].

In the late 1970s, Smalltalk was created with the goal of creating a programming utility device for children, but became largely adopted by professional programmers because of the cost of the systems on which it then ran.

2.1.2. Spreadsheets

Prior to 1979, if a manager wanted to include calculated information derived from the company's computer systems in a report they would have to request and negotiate with the IT department and it could take weeks! Then VisiCalc was released and everything changed. MBAs and others who could manipulate spreadsheets became king, and what-if and scenarios became a mainstay of corporate decision-making.

Today, classical spreadsheets are becoming less useful as data becomes too large and too dynamic to usefully capture in spreadsheets.

And only a small fraction of the population can use anything near the full power of a spreadsheet!