CityHero
Improving cities infrastructures

Andreas Scholz
Alina Uhrmann
Maximilian Meyer
18.07.2017
ABSTRACT.

Cities are the places where many humans spend the majority of their lives. There is oftentimes little space, and many crowded places while trains and buses have to transport tens of thousands of people everyday. This infrastructure is exposed to a great deal of strain and use over time. Every once in a while, something breaks, and the city has to do its best to fix the problem. This paper proposes a system, employing human computation to rapidly spot the damage and fix it as fast and as efficiently as possible.
1 Introduction

At Munich subway stations, there are more than 770 automatic escalators and 170 elevators [1]. Additionally there are thousands of park-benches, bus stops and other infrastructure that is being ground down by use and time. The city is conducting huge efforts to maintain this infrastructure over the years, but it is in many ways greatly inefficient. What if human computation could be used to mitigate some of the costs and time it takes to do these repairs? That is where CityHero comes into play. CityHero is a maintenance and IoT system, that can be used to find and repair damaged infrastructure within a city. It is designed to help make maintenance and repair works more efficient, less time and resource consuming and generally better by bringing together all the relevant actors and optimizing their workflow.

![Figure 1.1: Flow of information within CityHero](image)

**How is damaged infrastructure spotted?** As can be seen in 1.1, CityHero uses two different methods to spot damaged infrastructure. On the one hand side we have got active infrastructure. This would e.g. be automated escalators or elevators. This kind of infrastructure is either already or should be equipped with sensors, that let control know if an error occurs, or if something is not working correctly. On the other side, there is the more difficult case of inactive infrastructure, like for example park benches or bus stops. If damage occurs here, management would usually not immediately know. CityHero offers the CityHero App though, which is employing human computation in order to let users of the app submit damage reports. This way even completely non-technical failures can be, with sufficient permeation throughout the public, noticed quickly.
Damage spotted. What now? As can also be seen in 1.1, once the damage has been reported by either an app user or one of the active sensors, the city can then dispatch its repair crews efficiently and quickly. This gives the advantage of always being informed of the state of infrastructure throughout the city and is at the same time a big step to having a modern city in the sense of IoT.

1.1 Related Information on the Topic Field

Shin et al. [2] looked at Korea’s ubiquitous city development cases. The study analyzes ubiquitous computing by illustrating current ongoing projects in order to give socio-technical insights on the nature and scope of IoT and ubiquitous computing. The findings of this analysis provide a insight toward a future information infrastructure in the context of changing sociotechnical dynamics. The study heavily focuses on the human aspects of technical and political issues within cities. It is therefore relevant to our system as well. We must not forget, that ubiquitous computing must stay within ethical and moral limits. Changing technical cityscape may have negative side effects regarding democracy and ecological sustainability.

Ma et al. [3] have looked at the safety issues arising from the use of smart services and IoT platforms within cities. Their demo simulates parts of New York City and depict how CityGuard, their system, identifies unsafe actions and thus helps to prevent the city from safety hazards. Their demo is related to our project even though they look at different problems. They test their system for various performance metrics. We could in the future also check CityHero like this.

1.2 Purpose of the HC System

Cities are very complex organisms. There is bureaucracy slowing down repair works and everybody has experienced something being broken for weeks or even months at some point. At the same time, many stakeholders and rights holders are involved in the process which complicates decision-making [4] and bogs down the process. The purpose of this system on the one hand side is to facilitate repair works on the cities infrastructure by being able to assign and dispatch repair crews and on the other side to spot present damages by letting citizens use the CityHero application on their smart phones. Another purpose of our system is to heighten efficiency with the help of human computation and also decrease costs by consolidating different inefficient systems into one entity. We believe, that our system is able to significantly impact maintenance costs in the long run. The reason being, that the city management is, after a couple of years, at a point where they will be able to reliably predict costs for future years.

1.3 Human Contribution to the System

Why do we need humans within our system? The reason is, that nowadays image recognition and automated video analysis is not advanced enough to make abstract connections or even recognize simple content. Humans can easily spot an image containing “a horse within a forest”. A computer program still has great problems even recognizing even such simple scenes.
Humans, in our system, do the recognition work, that computers cannot do yet. Especially in the case of inactive infrastructure, like park benches, bus stops etc. our system needs information from the app in order to know about existing damages.
2 FUNCTIONALITY OF THE CITYHERO SYSTEM

2.1 FUNCTIONALITY AS SEEN BY A USER

Which functionalities does the user see? Most important of all, there is our CityHero-App. It is used to report damages throughout the city. Whenever a citizen spots a damaged piece of infrastructure, our app gives them the possibility to enter text, and/or take a photo with a geolocation tag. By using the app, the user may soon see an improvement of overall city cleanliness and functionality.

2.2 FUNCTIONALITY AS SEEN BY A STAKEHOLDER

Which functionality can the stakeholder take advantage of? Who are our stakeholders? In cities, there are usually many stakeholders involved in public property and its management. Sometimes every park in a city has its own management and repair service. For our system, we will assume, that there is cooperation between these entities. Our city then has a number of possibilities.

1. Receiving damage reports from sensors: The city has the ability to receive problem reports from the active infrastructure via sensors. E.g. elevators will send an automated report if they stop working.

2. Receiving damage reports from the app: Citizens using the app can also submit damage reports and have the city review it. The stakeholders then have the ability to process the information and take advantage of the human computation the citizens do.

3. Managing information influx and overview: The stakeholders will, once the system is rolled out within a city, constantly receive information about broken infrastructure. Our system now gives them the ability to manage this information and decide when and how to tackle the problems. This kind of management leads to efficiency. Our system gives an overview over total damage count and upcoming maintenance work to the shareholders.

4. Repair & maintenance: After a couple of months of deployment, the CityHero system will have collected enough information to reliably forecast the monthly or annual number of maintenance occurrences. Finally the cities repair crews can be effectively managed and dispatched to where they are needed the most.

INCENTIVIZATION CONCEPT

We hypothesise, that every human wants the city he lives in to be a nicer place. So why use our system? For two reasons. Firstly, humans have an innate need to feel good about themselves and help their community. This altruism will be harnessed by our system in order to convince people to use it. Secondly, if the city is healthy and in a good state, the user himself will feel better about it and more at home. For that reason, users are going to want to take advantage of our system, in this case its app.
Users will also be able to earn achievements, that will make them feel better about themselves. If, for example, a damage gets repaired, that has been reported by this user, a achievement called "First time city maintenance" could be issued. These achievements could then also be shared on social media. There could be all kinds of achievements to keep the users engaged and have them feel connected to the city. The achievements mainly serve to highlight the feeling of having done something good for the city and fellow humans.
3 SYSTEM DESIGN AND UI ELEMENTS

3.1 SYSTEM ARCHITECTURE

The CityHero ecosystem consists of three high level components as can be seen in figure 3.1. The users, the cities and our own CityHero infrastructure. The users report issues with the city's infrastructure via the CityHero app. The reported data is aggregated by us and the conclusions drawn from it are passed on to the cities. The city will act upon the recommendations and in return provide feedback whether or not the conclusions we have drawn from the available data were correct. We can use this information to further improve our aggregation and recommendation algorithms and to reward the users when they reported something of value. All the interactions between these three entities take place via APIs offered by the CityHero infrastructure. More detailed information on the technologies involved can be found in chapter 3.3.

![Figure 3.1: CityHero Component Diagram](image)

3.2 DATA AGGREGATION

As with any human computation system, collecting data is useless without an appropriate method to aggregate this data and draw conclusions from it. However, to design such an algorithm it is necessary to know which kind of data is collected as well as how many users the application is likely to attract.

For our CityHero HC system the answer to which kind of data we collect is very straightforward. By requiring certain information during sign up we can make sure that we have a minimal set of information for each user. This information includes, but is not necessarily restricted to user name, city and email address. Optional data fields could contain items such as name, last name, date of birth, gender, phone number, home address, Facebook or Google+ accounts and others. Deciding upon which of this information should be required upon sign
up is - apart from a data science view point - also a philosophical question and shall thus not be answered in this section. However, we can conclude that there will be a certain minimal set of information available for each user, as well as optional data that will only be available for some users. This is something which has to be considered during data aggregation and decision making. The amount of data that will be collected is not as easy to anticipate. Assuming that we start CityHero in one of the major cities in Germany, we can expect the population of that city to be the upper limit of users we can attract. Considering that our target group will likely mostly contain people from 15 to 64 years of age, this decreases the upper bound to approximately 66% of the population [5]. For simplicities sake we will assume that any one of those people has a mobile phone capable of running the CityHero application. In the best case scenario we would reach all of these people via advertising and word of mouth. Assuming a strong conversion rate of 2% [6], this would leave us with 0,0132% of all citizens. According to studies the amount of active users within any single distributed system ranges from 1% [7] up to 10% [8], so that we can expect to have at most 0,00132% of all citizens as users for any given city. For the city of Munich with a population of 1.450.381 [9], this would translate to about 1.915 active users in the best case scenario.

It is also worth mentioning that a human computation system such as CityHero should be able to adapt over time as more information becomes available. CityHero might expand to other cities and countries, which could drastically increase the amount of data available for aggregation and analysis. All this lead us to believe that we need a way to aggregate the data collected by our system in a way that provides solid results even with a small user base, but is extensible enough to employ more sophisticated aggregation methods once enough data is available. It should also be adaptable to the amount of information available per user.

To this aim we suggest a four step process to aggregate and evaluate the data collected by CityHero and to continuously validate and improve the predictions based off the data.

1. **Initialization**: Computing a trust score \( ts \) for each user based on the mandatory \( m \) and optional \( o \) information available for that user.

2. **Aggregation**: Based on \( n \) individual users reporting the same issue, each associated with their own trust score, compute an overall certainty score \( cs \) of whether the reported problem is real.

3. **Decision**: If the certainty score is above the acceptance threshold \( at \), consider the reported problem to be real, otherwise reject it.

4. **Verification & Adjustment**: Based on the feedback of the repair crews whether the reported issue was indeed real, adjust the overall acceptance threshold and the trust scores of the users having reported this issue accordingly.

We purposely formulated this process very generally to allow for different implementations depending on the amount of available data. A very simple algorithm implementing these steps is shown in figure 3.2, subfigures (a)-(d). Subfigure (a) shows the initialization of the
trust score and the acceptance threshold. The values chosen for $ts$ and $at$ make sure, that at least three reports by new users are required to accept a certain issue. These numbers are chosen via a best guess approach and will be modified by the algorithm later on to increase precision. Subfigure (b) shows the computation of the certainty score for a single issue. Subfigure (C) depicts when a specific issue is accepted as such and when it is rejected. Finally, subfigure (d) depicts the adjustment for each users trust score, who have reported an issue correctly or incorrectly as well as the adjustment of the acceptance threshold. In this simple algorithm the acceptance threshold is supposed to be adjusted manually by finding a balance between false positives and false negatives. We feel that this is the best approach considering the small amount of data available initially.

Later on however, with enough data, machine learning approaches will be feasible. Instead of initializing each user with a fixed trust score, a sophisticated algorithm could take the mandatory and optional data available for each individual user as an input vector to compute a more realistic trust score. This could be useful to find correlations between e.g. the age of a user and their trustworthiness. Adjusting a users trust score can also be more individualized. With enough reports available by a single user it is possible to adjust the boost or lowering of their trust score based on their past performance. For example, the trust score of a user who has a flawless reporting record in the past should not be lowered as much in case of an accidental wrong report as that of a user who repeatedly made false reports. Finally, the acceptance threshold could be adjusted according to data based metrics to find the optimum for each city. Some cities might have a higher threshold for false positives than others and this could be taken into account when adjusting the acceptance threshold.

As has been shown, there is a lot of potential in aggregating and evaluating the available data to make the best predictions possible. Considering the low volume of data available when starting out we think it is beneficial to start with a simple algorithm which can become more sophisticated as the number of cities and therefore the number of users and the available data grows.

3.3 Technologies

The CityHero ecosystem consists of three high level components as can be seen in figure 3.1. It has to provide a way for users to report damages with their mobile phones, process this data to extract valuable information and provide this information to the cities.

User Technologies To reach as many users as possible, it is important to support the most widely used mobile phone operating systems(OS). Android, iOS and Windows Phone together have a market share of 99.9% [10]. However, it is very resource consuming to create native apps for each of these OS. We will therefore use HTML5 to create a website which can be accessed by each of these OS. This website will look and feel like a native app on each of these devices and can be downloaded like any other app from the according app store. This is possible due to the device APIs that are provided by each of the OS. These APIs let us access features of the mobile phone, like camera and geo location via HTML5 [11][12][13][14]. This has the benefit of having to maintain only one code base instead of three.
\[ ts_{n_i} = 0.5, \quad at_{city} = 1.5 \]  \tag{3.1}

(a) Initialization of the trust score for each user and the acceptance threshold.

Only done once.

\[ cs_{issue} = \sum_{i=1}^{n_{issue}} ts_{n_i} \]  \tag{3.2}

(b) Computation of the certainty score.

\[
\begin{align*}
    cs & \geq at :: accept \\
    cs & < at :: reject \\
\end{align*}
\]  \tag{3.3}

(c) Condition of acceptance or rejection of a reported issue.

\[
\begin{align*}
    at & = at' \\
    ts & = \left| ts - \frac{ts}{10} \right| (min = 0.1) \\
    ts & = \left| ts + \frac{ts}{10} \right| (max = 1.0)
\end{align*}
\]

(d) Adjusting the trust score of each user having reported a verified correct or incorrect issue.

Figure 3.2: A simple algorithm to aggregate the users data.

**CityHero Technologies** Essentially, we have to provide three functionalities. Authentication, Computation and Storage. For authentication (with a user or city), we will provide a dedicated authentication server running OAuth 2.0[15] and an OpenID[16] instances, for both of which open source implementations are available. This will provide sufficient security for interacting with our application, as well as provide single sign on (SSO) capabilities. Our computation, data aggregation and interaction will be handled by our application server. A standard Unix Server running Debian 8 will suffice for these tasks. If the amount of data needing to be processed grows we can scale the computation to multiple such servers. Finally, storage will be handled by a dedicated database server. Depending on the final implementation of our CityHero application this can be either a relational or document based database.

**City Technologies** We assume that the electronic infrastructure of cities varies greatly. Instead of adapting to each city individually we will therefore provide an API for our CityHero application which can be accessed by the cities to query for information relevant to them. This API will also allow cities to report back feedback on the accuracy of our predictions. This API can for example be implemented via REST[17] or GraphQL[18], both of which offer
support for a whole range of software systems.

3.4 USER INTERFACE

The User interface has gone through three main stages of design. At first a paper prototype was drawn out and stylized on paper with the main purpose of gathering live feedback from potential users. Figure 3.3a and figure 3.3b show the first drafts on paper. Second a digital wire frame was built and optimized regarding the actual resolution and size of a smart phone. It was built with Evolus Pencil wire frame tool. At last the color scheme for the app was iteratively developed and evaluated. The Appendix shows the main design candidates with a short description on their strengths and weaknesses. The final iteration can be seen in figures 3.4a, 3.4b, 3.5a and 3.5b. The particular design and colors were chosen for clear readability, ease of use and colors to highlight the main features of CityHero.

After installing the app on their phone the user is asked to log in or sign in via a social network or by using an e-mail account as seen in figure 3.4a. Our app focuses on minimalistic and easy to use design, which means a user has to have a minimal amount of obstacles getting to the core functionality of the app, the reporting tool. That is why the screen shown in figure 3.4b is the first screen seen by a logged in user when opening the app. The reporting tool shown in figure 3.4b offers different methods capturing the core information about an infrastructural issue. Users can submit their location, standing next to the issue. A photograph picturing the issue can be made using the camera. And lastly a textual description of the problem can be typed by the user as well. As suggested by Arpita Ghosh in [19] our system is gamified using a badge reward system. Submitting a report instantly gets gratified by a badge displayed on the screen, providing an incentive for continuous participation in the system. Figure 3.5a shows the screen after a user has contributed a report. Tapping anywhere on the screen lets the user return to the reporting tool.

All badges earned are displayed in the gallery shown in figure 3.5b and can be shared across the social network you logged in with. There are different kinds of badges, though: Earned badges, not earned badges with a description on how to obtain them and Easter egg badges without a description. Not yet earned badges are displayed as silhouettes in the gallery. After the reported issue has been verified the creator of the report gets a push-notification, further involving the user in the system. Tapping on the push notification opens the app and displays another congratulating screen, intensifying the rewarding feeling of success. Optionally a user can create and fill out a profile page. If a user has any trouble using our product he can always tap on the contact us/support icon in the menu.

A currently developed extension is the history function to enable a user browse through the history of his or her reports, which are updated based on whether their report was verified, how the repair team accessed the real damage in contrast to the one described in the report. A prototype screen can be seen in figure 3.6.
4 System Evaluation and Success Criteria

4.1 Limitation of the System

Our system has also limitations in its use. For instance, our human computation system relies on its users for damage reports. Without users, or even sufficient amounts of users, the whole system collapses as it lacks information on where to send any repair crews. Also, without any precautions in place, the human computation system is prone to acknowledging false reports and users who constantly and collectively provide unreliable reports. Different motivations can be at play here: ignorance, personal gain of social status the "easy way" or just for fun as described in [20].

Our proposed system architecture aims to include every citizen who has a smart phone with Internet access, which gives us plenty potential reporters who are - unfortunately - in no way trained to detect issues in the infrastructure. This circumstance means that issues only identifiable by professionals will most likely be missed. Another problem arising from that could be the unskilled reporters who are not able to sufficiently describe the issue they are reporting, leading to badly described and documented reports which are harder to analyze.
4.2 EVALUATION- AND SUCCESS CRITERIA

In order to evaluate the system there need to be objective criteria that help us to measure success objectively. The initial intuition is to look at the number of registered users in the app. More people using the system indicates people spreading the word about the app and embrace the concept of CityHero. But this is not enough. It is more important to measure if the goal of the human computation system is approached. Did our system improve the infrastructure of the city? Another approach would be to test the overall satisfaction of the citizens regarding their experience using the public infrastructure or looking at the number of reports submitted with and without CityHero. The problem here is that increased report quantities can also mean that more small issues are being reported too. At last we can focus on the work that has been done by the dispatched repair crews. Are the repair teams deployed more effi-
(a) CityHero player earning a badge

(b) Earned and earnable badges in the gallery. Earned badges can be shared across social media.

Figure 3.5: CityHero colored Prototype: Gallery and Badges

ciently? In order to measure the divergence we can employ a w-test to test the distributional difference between our two cases as showcased in [21]. Our success criteria is then the comparison between the efficiency of the repair crews before the system got implemented versus the current level of efficiency.
Figure 3.6: Report history of a user.
Favorite reports are on top, followed by other reports sorted chronologically.

5 Future Works

5.1 Possible extensions of the HC system

Our system is easy to extend to different parts of city life besides only managing city repair crews. Emergencies can be tracked with the app by extending the report function with the flag "emergency". Emergencies would be then submitted directly to the police when facing robberies or violence or transmitted to medics when there is a medical issue at hand. It is necessary to prevent abuse of this feature by holding each user accountable for inappropriate use coming from their devices. For better visualization an interactive map of the city with pins showing the reports in your area can be included to the core app. With some gathered data we can also predict which areas of the city are prone to damages and how this affects the housing market to help people settle in a friendly and save neighbourhood. To maximize
overall service it can be investigated if the outsourced repair crews prioritize certain reports over others when facing multiple requests for repairs at once. Optimizing the prioritization strategies would then directly benefit the citizens. Not only cities would greatly benefit from our system. Public places like universities can use our human computation system to track the broken or used equipment of the facilities like beamers, light bulbs and others. Based on the reports statistics of the most used items can be created which highlight the most fragile or worn out pieces. Cleaning of lavatories, garbage collection supervision can also be approached with our system. Generally speaking, our human computation system helps solving problems using people to monitor the status quo and report any unusual or malicious circumstance.

5.2 Thoughts on interaction with other HC system

CityHero is very well suited to join forces with "stress and transport" from Team Obazda in the human computation lecture of summer 2017 held by professor Bry at university of Munich. Providing important data on especially save and functional neighbourhoods versus the not save and demolished ones CityHero can help guide the user with less stress through the streets.
Appendix

Figure 5.1: CityHero original prototypes: digitalized paper prototypes. Were rejected as they seem too cold and boring for users. Have optimal contrast, though.
Figure 5.2: CityHero original prototypes: Design 1. Has too low contrast, hardly readable. Users loved the warm and inviting feeling of colors.
Figure 5.3: CityHero original prototypes: Design 2. Second favourite overall.

Users loved the playful colors.

Lacks contrast on beamer screen.
Users loved the orange getting their attention, contrast is also good. Blue hues were too dark and scary, beamer interprets the blue hues as black.
Very good contrast, alerting colors for emergencies. Users felt design was too aggressive.
Figure 5.6: CityHero original prototypes: Design 5.

Good contrast.

Added blue and lighter red did not help enough to reduce aggressive feeling.
REFERENCES


