

# Detection of Runtime Conflicts among Services in Smart Cities

M. Ma <sup>\*</sup>, S. Masud Preum <sup>\*</sup>, W. Tärneberg <sup>†</sup>, M. Ahmed <sup>\*</sup>, M. Ruiters <sup>\*</sup>, and J. Stankovic <sup>\*</sup>

<sup>\*</sup> Department of Computer Science  
University of Virginia,  
Charlottesville, VA 22903, USA

Email: {meiyi, preum, mohsin.ahmed, mcr5fh, stankovic}@virginia.edu

<sup>†</sup> Department of Electrical and Information Technology  
Lund University,  
Lund, Sweden

Email: william.tarneberg@eit.lth.se

**Abstract**—The populations of large cities around the world are growing rapidly. Cities are beginning to address this problem by implementing significant sensing and actuation infrastructure and building services on this infrastructure. However, as the density of sensing and actuation increases and as the complexities of services grow there is an increasing potential for conflicts across Smart City services. These conflicts can cause unsafe situations and disrupt the benefits that the services were originally intended to provide. Although some of the conflicts can be detected and avoided during designing the services, many can still occur unpredictably during runtime. This paper carefully defines and enumerates the main issues regarding the detection and resolution of runtime conflicts in smart cities. In particular, it focuses on conflicts that arise across services. This issue is becoming more and more important as Smart City designs attempt to integrate services from different domains (transportation, energy, public safety, emergency, medical, and many others). Research challenges are identified and then addressed that deal with uncertainty, dynamism, real-time, mobility and spatio-temporal availability, duration and scale of effect, efficiency, and ownership. A watchdog architecture is also described that oversees the services operating in a Smart City. This watchdog solution detects and resolves conflicts, it learns and adapts, and it provides additional inputs to decision making aspects of services. Using data from a Smart City dataset, an emulated set of services and activities using those services are created to perform a conflict analysis. A second analysis hypothesizes 41 future services across 5 domains. Both of these evaluations demonstrate the high probability of conflicts in smart cities of the future.

**Index Terms**—Smart Cities, Conflict Detection, Conflict Resolution, Services, Watchdog Architecture

## I. INTRODUCTION

Urbanisation is not a new phenomena, but has never before proceeded so rapidly. Over the last century human migration and aspiration have concentrated to its cities. The reliable availability to diversity of employment, public services, and housing is strong incentive that drives this change in human habitation. Further, cities no longer serve their population with a set of discrete services. Increasingly, public services are seamlessly intertwined and support each other, from transportation to health-care to utilities.

With the arrival of technological tools such Internet of Things (IoT), Big Data, Cloud computing products [1], and Crowd Sourcing platforms, cities are becoming increasingly able to monitor the state of their infrastructure, services, and populace, cost effectively and at scale. With a connected

populace and infrastructure, cities are also able to dynamically act on changes based on the observations it makes with increased accuracy. A city that employs such technologies to mitigate the strains of urbanisation, to improve the quality of life for its inhabitants, and the competitiveness of its economy is commonly referred to as a Smart City. There are a number of cities that are already embracing the notion of a Smart City, such as the city of Santander in Spain [2].

One key open problem is that with many services operating simultaneously, conflicts will arise. Conflicts have both an immediate effect on human life, as well as long term secondary/tertiary effects. In the complex system-of-systems that is a Smart City, services will come into conflict when contending over the same resources, incurring opposing actions, and when having contradictory or conflicting objectives. These conflicts are both institutional and technical, and have to be resolved holistically. Finding and classifying conflicts are non-trivial but crucial to the operation of a Smart City due to (i) the scale of a Smart City system, (ii) the diversity in services, and (iii) the wide range of ways the services interact with the city. While some of these conflicts can be detected during the design phase of services, many conflicts can occur unpredictably at runtime, i.e., when the implementation phase is over and the services are operating simultaneously. Detecting runtime conflicts is significantly more challenging than detecting design time conflicts, as runtime conflicts involve a higher degree of uncertainty. Once a runtime conflict is identified, resolving it often involves a compromise, such trade offs have both a technical and administrative component.

This paper primarily explores the nature of the runtime conflicts that arise in a Smart City and how they can be mitigated through a watchdog architecture (Figure 1). The primary contributions of this paper are as follows:

- The enumeration of Smart City services characteristics.
- The classification of conflicts.
- The design of a watchdog architecture to detect and resolve these conflicts.
- An evaluation of conflict analysis that demonstrates the high probability of conflicts using actual data from a Smart City.
- An evaluation of conflicts using 41 future services that shows that the high probability of conflicts can be ex-

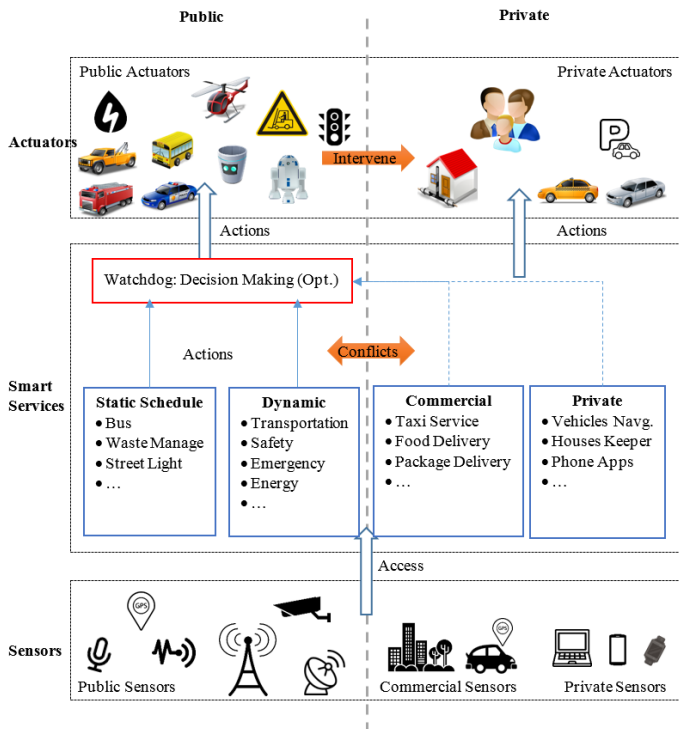


Fig. 1: The Integrated System of Smart Cities

pected to grow in the future and that many conflicts can only be resolved at run time.

## II. SMART CITY SERVICES

A Smart City is a system of systems, where each system represents a specific domain (e.g., transportation, public safety, utility, emergency, environment, city planning and operations) and each domain consists of a set of services. For example, the public safety domain may include police patrolling services, traffic violation control services, and road accident management services, etc. Similarly, the transportation domain includes public transport services, road work services, etc. Each service performs a set of functions to fulfill an objective, e.g., a traffic violation control service penalizes drivers for speeding. The functions may be triggered by an event (e.g., a traffic violation) or scheduled statically (e.g., turning off street lights at dawn to save energy). Functions may produce a set of effects upon completion, e.g., blocking a lane for road work. Effects that are directly actuated by a service are primary effects. Effects that are the outcomes of a primary effect are characterized as secondary effects. Thus a single action can create a chain of subsequent effects.

Potential services are described from different domains in Table I. Although not exhaustive, typical services that covers several major functions of a Smart City are included<sup>1</sup>. How such essential services frequently conflict with each

<sup>1</sup>Most of these services either exist already in some cities or are going to be implemented in the near future.

other when they run simultaneously will be demonstrated in (Section V-B).

### A. Characteristics of Smart City Services

The characteristics of a Smart City service describe how it interacts with the city's resources and other services, how it affects the environment and people, and what requirements it imposes on the infrastructure it employs, such as sensors and actuators. Below some fundamental characteristics of Smart City services and their interaction with each other are introduced.

**Uncertainty:** It refers to unforeseen events or processes that can not be accounted for by the services ahead of time or are unknown to the service but affect its performance. Uncertainty can occur in the different layers of a service: sensing layer (a pollution sensor attached with a vehicle moved from its expected location), communication layer (network failure), or actuation layer (control valve only opens partially). Uncertainty usually arises in the flow of information and/or resource availability. Uncertainties can cause change(s) in the course of action. This characteristic emphasizes the need to design an integrated service platform.

**Dynamism:** Although some services of a Smart City can operate with a static schedule (e.g., sending out a garbage truck every morning), a major portion of services function dynamically. For example, public transport service schedules bus routes and frequency based on demand: more buses when there is a festival/game/concert. Such dynamic operation of a service can create complexity if the service shares resources (sensors/actuators/data) with other service(s).

**Real-time:** In a Smart City, services frequently rely on real-time information for operational decision making. One of the most common Smart City service is monitoring crime-prone areas through surveillance cameras and mitigating risky incidents. It relies heavily on real-time video feed.

**Mobility/Spatio-temporal availability:** Often services involve mobility in terms of sensing and actuation, e.g., police patrol service and garbage collection service. This poses a new set of challenges, including, coverage, redundancy, operational cost, scheduling, communication between the control and physical layers. Also, mobility contributes to uncertainty, and the degree of dynamism.

**Duration and scale of effect:** Any function performed by a service results in effects or a chain of effects into the future. These effects vary in duration. For example, blocking a road can affect traffic for a long time. Some services require large and/or lengthy actuation to make small changes, while others require small and/or short actuation to make significant changes. This complicates the control mechanism. In addition, the notion of duration and scale of effect can lead to uncertainty, real-time feedback, and resource constraint.

**Efficiency:** Service behavior is determined by the targeted efficiency. Efficiency can be measured as a function of resources, cost, and time. Targeted efficiency can control the number of sensors and actuators used in a function of a service, e.g., increasing the number of buses to meet public demand

TABLE I: Future Services in Smart Cities

ID	Services	Domain	Description
1	Street Lights Control [3]	Environment	It controls level of illumination in city streets by 1) turning on/ turning off street lights and 2) adjusting brightness of street lights according to ambiance.
2	Street Robots Management	Environment	Robots on the street sense different environment states (e.g., weather, light, pollution level) and aid a passer by if necessary (e.g., kid, disabled people).
3	Waste Management [1]	Environment	It performs waste collection, disposal, recycling, and recovery. It sends out garbage collection trucks regularly and extra ones when the containers are over 2/3 full.
4	Delivery Management	Environment	It maintains a dynamic schedule of package delivery trucks based on real-time demand.
5	VIP Delivery Management	Environment	It dispatches drones to carry packages to customers.
6	Air Pollution Control [2]	Environment	When air pollution is detected to cross the safety threshold, it will 1) send out personalized sms to citizens based on location and physiological state, 2) post messages on street screen, and 3) suggest authorities to determining the cause of pollution and reduce vehicles on the streets.
7	Noise Pollution Control [2]	Environment	If noise pollution is detected to cross the safety threshold, it will 1) send out sms to citizens, 2) post messages on street screen, and 3) turn off public speakers/alarms.
8	Port Pollution Control	Environment	It detects potential pollution caused by incoming ships and stops polluting ones from coming to the port. When serious pollution is detected, it will send out sms to nearby ship and stop them from coming.
9	Culture Event Management [2]	Environment	It facilitates the diffusion of information about cultural activities and motivates people to be involved in them. It also helps to manage the facilities (parking, lights ) and around neighborhoods during cultural events.
10	Live News Gathering [4]	Environment	It manages navigation of autonomous news coverage vans and drones to breaking news sites for capturing video/photographs.
11	Adaptive Traffic Light	Transportation	It adjusts signal lights dynamically based on traffic density to maximize utilization and prevent traffic congestion.
12	Emergency Vehicle Monitor	Transportation	It monitors streets and adjusts traffic lights to minimize delay of emergency vehicles, e.g., police cars, ambulances, firetrucks.
13	Road Condition Monitor	Transportation	Upon detecting light snow/rain, it sends alarm to nearby drivers. Upon detecting heavy snow/rain/flood, it adjusts signal lights to block the road and reroute vehicles.
14	VIP Route Scheduling	Transportation	It reroutes regular vehicles off VIP routes and programs the shortest path for VIP cars.
15	Traffic on Special Events	Transportation	To accommodate visitors during games/concerts/other events, it blocks some streets and adjusts traffic signals on event days.
16	Road Work Service	Transportation	It manages road works and road side constructions. It reroutes vehicles to alternate paths when road work is going on.
17	Smart Parking [5]	Transportation	The system informs drivers about the number of available parking spaces in adjacent areas and gives direction to desired parking lot.
18	Bus Schedule Service [6]	Transportation	It manages the bus schedule both statically and dynamically. Based on passenger demand, it reduces bus interval/waiting time and directly sends extra buses to some bus stops.
19	Taxi Dispatch Service [6]	Transportation	In case of increase in demand, it sends extra taxis to corresponding event locations.
20	Traffic Violation Control	Public Safety	It pulls over vehicles for traffic violations.
21	Road Accident Management [4]	Public Safety	In case of a road accident it 1) notifies law enforcement services and GPS navigation services, 2) blocks roads temporarily, 3) sends message to vehicles and street screens, and 4) adjusts traffic lights to regulate traffic flows and prevent traffic jams.
22	Risky Area Monitor [6]	Public Safety	It co-ordinates sensors (i.e., cameras, street lights) and actuators for real-time monitoring of risky areas. Upon detecting any crime or police intervention, it alerts citizens to avoid such areas temporarily.
23	Raiding Crime Scenes	Public Safety	It conducts raiding operation in crime scenes/risky zones. During the raid, it can block roads without any prior notice if necessary.
24	Destroying Obsolete Structures	Public Safety	It blocks some roads temporarily when blowing up any obsolete structures: bridge/buildings. It informs residents in nearby areas ahead of time and blocks off nearby roads during the operation. Informing adjacent vehicles about the event involves uncertainty.
25	Potential Terrorist Attack Monitor	Public Safety	When it detects a potential terror threat / attack it 1) postpones operation of trains/public transports in concerned areas, 2) re-routes vehicles, and 3) often uses bio-chemical weapons against potential threat(s).
26	Surveillance Drone Management	Public Safety	It uses drones to monitor safety conditions and detects potential threats over streets and buildings.
27	Public Security	Public Safety	It helps public organizations and houses to protect citizens' goods and feeds real-time information to fire and police departments when detects an intrusion or theft.
28	Fire/Explosion Management	Emergency	It detects and automatically takes action based on the level of severity, such as: 1) detecting false alarms, 2) informing firefighters and ambulance, 3) blocking off nearby streets/buildings if necessary, 4) helping people to evacuate, and 5) co-ordinating rescue drones and robots.
29	Inclement Weather Alert	Emergency	It alerts and gives personalized advice on how to stay safe during emergency (storm, earthquake, tsunami,flood) through messages, suggest the car to stop upon detecting earthquake, manages street lights and other utilities safely.
30	Evacuation Aid	Emergency	It helps people evacuate during extreme emergencies. Specifically, it detects the location of people and sends rescue, map, message of instruction to the phones, and manages drones and robots for rescue service.
31	Automatic Health-Care Dispatch	Emergency	It provides 24/7 health care for patients. When it detects an emergent situation of patient, it will sends an ambulance or helicopter to pick up the patients and send them to the most suitable hospital.
32	Ambulance Management	Emergency	It sends ambulance to help patients and send them to the nearest hospital when someone calls an ambulance.
33	Water Pipe Monitor	Emergency	Upon detecting pipe leakage, it turns off the water flow and sends service crew. It blocks street around if necessary (e.g. manhole concerned).
34	Gas Pipe Monitor	Emergency	It monitors gas pipe, and alerts and evacuates people around based on different degree of severity, cuts off electrical utilities if necessary and sends service crew upon detecting the gas leakage.
35	Electricity Monitor	Emergency	It starts the back-up generator, and sends crew upon detecting any technical fault. It also sends alert to adjacent people and take proper intervention.
36	Gun-Shot Detection	Emergency	It alerts nearby patrol police immediately and sends messages to people in the same neighborhood. It also analyzes surveillance (if any) to find potential suspects.
37	Network Error Detection	Emergency	When detects network error, it informs people immediately. After network error, it exams all services if they are functioning properly or not.
38	Sleep Mode for City	Emergency	It turns the city into sleep mode gradually if serious emergency arises (utility breakdown, network failure etc.) while guaranteeing the basic functionality of city.
39	Water Usage Monitor [2]	Energy	It monitors water usage and turns off water flow for a short period if excessive water is consumed continuously for a long duration.
40	Energy Usage Monitor [2]	Energy	It turns off electrical devices at idle hours to save power.
41	Solar Energy Generation Optimization	Energy	When solar energy is available, it turns machines to solar energy mode.

on weekdays. Maximizing efficiency for one service can often lead to resource constraints for another service if the two services share any resource. Thus it poses an optimization problem with constraints on resources and operational costs.

*Ownership:* A service can be private, public, or commercial in terms of ownership. The degree of interaction and information flow between services with different ownerships can vary according to service design, and city policies.

Although not completely, characteristics above outline the potential **complexity** of Smart City services. Moreover, the characteristics play a vital role in creating the context of potential **conflicts** among the services as described in Section III. It is acknowledged that these characteristics affect one another and cannot be quantified on their own. For example, mobility and efficiency affect uncertainty.

### B. Integration of Smart City Services

Service integration is integral in the context of a Smart City for the following reasons.

First, a service might frequently interact with other services from the same domain as well as from other domains. For example, when there is a road accident, both the road accident service and the emergency dispatch service usually act together to address the situation. Service interaction can occur at different degrees based on the corresponding situation. To make such interactions functional, efficient, and frictionless, services must be integrated. The integration process must start from as early as the design phase of the services.

Second, some scenarios in a Smart City require services to share resources both in sensing and actuation layers. For instance, the same set of trucks can be shared for collecting garbage and carrying air pollution sensors. An integrated service platform that is aware of the demands, constraints, and objectives of each service, can contribute to efficient scheduling of shared resources. Thus service integration is vital for quality of services (QoS) and operational efficiency.

Third, services are often correlated with each other in terms of sequence of operations, i.e., output of one service is the input to another service. For example, a road accident can cause traffic delay. If a GPS enabled navigation service (e.g., Google maps) is not aware of the accident, it may yield an erroneous estimate of the route time. Services must be integrated to ensure proper data flow among them.

## III. CONFLICTS IN SERVICES

This section describes how the different characteristics of Smart City services contribute to conflict. Specifically, it presents potential sources of conflicts and enumerates various types of conflicts.

Broadly speaking, conflicts in services arise when the actions from two services can not be performed together without adverse effects. Conflicts can occur at devices, in the environment or upon people. Below a list of potential sources of conflicts is presented. Although the list is not necessarily complete, it covers the potential conflicts which are empirically observed from the services listed in Table I.

### A. Device Conflict

One of the most common sources of conflicts is a shared device in the actuation layer and/or sensing layer. For example, Service 17 (S17) from Table I sends vehicles to the parking garage, where S9 does not allow any outside vehicles into the garage on a special event day. Device conflicts in smart cities can occur on stationary devices (e.g., street lights, traffic signals, message boards, etc.) as well as on mobile devices (e.g., vehicles, drones and robots). Comparing with the stationary devices, conflicts on mobile devices are more difficult to detect and resolve. Without a watchdog layer, the conflict between these two services could not be detected until it really happens.

### B. Environment Conflict

Besides the direct conflicts on shared devices, services are also prone to indirect conflicts caused by contrast effects on the environment. For example, S6 and S18 from Table I do not have conflicts on a device because they do not share any device. However, when the air quality is bad, S6 might want to limit the number of vehicles on the road. In contrast, S18 can schedule more buses for passengers at the same time to meet public demand given it is a busy day. As a result, S6 and S18 will cause environment conflict because of their contrary effects on air pollution level.

### C. Human Conflict

Humans are at the center of Smart City services and cause conflict in several ways. First, Smart City services include decision optimization aids, but often also rely on humans working for the city to make final decisions. When 10s or even 100s of humans are making decisions across many service domains, the possibility of conflicts is significant. Because humans are subjective and they tend to make the decision based on personal bias/priority/incentive. For example, in case of co-occurrence of a severe fire and several emergency police calls, decisions on how to dispatch the fire service, police, and ambulances might vary widely among the people who are in charge of these services. Second, human conflicts are caused by contrary effects of environment on human physiology. For example, a decision to temporarily allow greater pollution in an area of the city to ease major traffic congestion can adversely affect asthmatic people in that area. This is a conflict between human physiology and environmental effect. Third, human conflicts can also occur from adverse effects of services on a single person and group.

### D. Typology of Conflicts

While the above list describes under what circumstances conflicts may occur, the below list categorizes various types of conflicts.

1) *Opposite Conflict:* An opposite conflict is caused by opposite actions on the same device, on the environment, or on the human taken by different services. For example, traffic congestion service wants to turn the traffic lights to red while the emergency service wants to turn them to green so that an ambulance can pass quickly.

2) *Numeric Conflict*: A numeric conflict is caused when actions from different services request different values for one or more parameters of a shared resource. For example, when a storm is coming, the water level monitor service wants to set the water level at  $X$ , while the ship management service needs to set the water level to  $Y$  to ensure it is high enough for anchoring the incoming ship. Here, conflict will occur if  $X$  is unequal to  $Y$ .

3) *Duration Conflict*: It occurs when two actions are similar and their start time is the same but the duration is different. For example, an accident detection service needs to block the whole road for 20 minutes to deal with an accident, while the traffic congestion service allows the road to be blocked only for 10 minutes.

4) *Completeness Conflict*: A completeness conflict refers to the situation when multiple actions are taken by one service to complete a task, but at least one of the actions is affected by the other services, and hence, that service can not complete the task. For example, a water pipe monitor service blocks nearby roads and asks for a crew truck when it detects a serious leak. However, the truck is stuck in a traffic congestion (i.e., effect of traffic congestion service). In this case, the pipe monitor service can not finish the task.

#### E. Consequences for Conflicts

1) *Safety Issues*: Assume that the Smart City sensors detect a traffic accident, or a disturbance such as a fight or riot, or there is a gas or oil leak. These incidents may occur independently or even all at once. Actions taken by Smart City safety service upon detecting a unsafe condition might include one or all of the following: dispatch police, ambulance, and/or firefighters, adjust traffic lights to reroute traffic, and inform the public through displays and apps on smart phones. However, other services such as transportation may detect congestion caused by these types of unsafe events and reset the lights or display messages differently than needed by the safety service. For example, the safety service might activate lights to permit an ambulance a non-stop route while the transportation service sets red lights differently to minimize congestion.

In general, it is non-trivial to create a set of services that can predetermine all the possible ways that the services may interact in real-time and under all conditions. This includes conditions such as (i) failure of sensors/actuators, (ii) actions and their consequences are not instantaneous, (iii) occurrence of random events such as a funeral procession or an earthquake, or (iv) entities that are not controllable, e.g., cars not paying attention to the advice/information. This emphasizes the need to create a real-time conflict detection and resolution module that will compare the safety critical nature of conflicts and outcome of various conflict resolution schemes.

2) *Environment and Health Issues*: Given environmental monitoring services in smart cities, once pollution is sensed, they will find the causes of the pollution and take actions accordingly. For instance, S6, S7 and S8 can block off the street and limit vehicles, send warning messages to the chemical factories, or prevent ships from coming to the ports of cities.

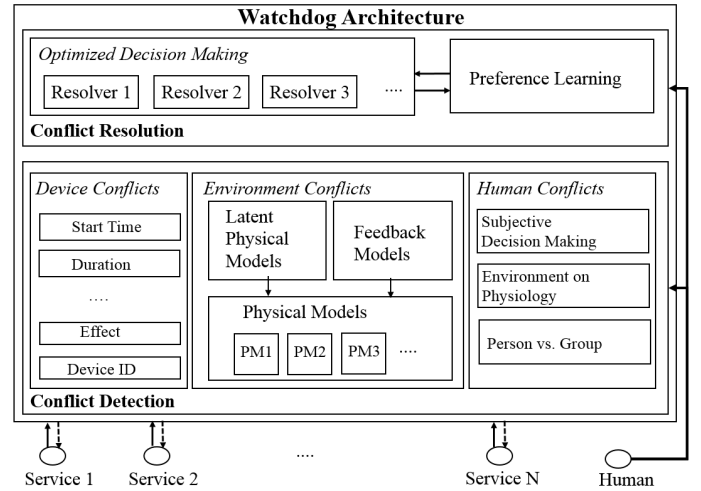


Fig. 2: The Watchdog Architecture

On the other hand, in order to make more profit, commercial services such as the ship management, taxi services and factories may ignore the warnings and even produce more pollutions gradually. Consequently, environment and human health will be impacted negatively. On the other hand, people also cannot close all the chemical factories and stop all the vehicles to reduce the pollution. Therefore, secondary or even tertiary effects of an action must be addressed while detecting and resolving conflicts. Conflict resolution system also needs to consider the severity of outcomes caused by conflicts, as the outcome can vary widely.

#### IV. WATCHDOG ARCHITECTURE

A two-stage watchdog architecture to detect and resolve conflicts among services of smart cities is designed (Figure 2) in this section. In the watchdog architecture, all requested actions by services are first intercepted and analyzed for potential conflicts by Conflict Detection stage. If no conflict is detected, the service actions are permitted. If a conflict is detected, the conflict related information is passed to the Conflict Resolution stage.

##### A. Conflict Detection

As outlined in Section III, conflicts arise in three general sources: devices, environment, and humans. In our proposed architecture, separated modules are dedicated to detect conflicts originating from each of the three sources in parallel. Below initial designs of these dedicated modules are described.

1) *Device Conflict Detection*: The Device Conflict Detection module determines if two or more services want to set the same sensor(s) or actuator(s) in opposite or incompatible states, or with different numeric values simultaneously. Since direct conflicts are occurring on known devices this type of conflict is relatively easy to detect. A comprehensive device conflicts detection system was developed in Depsys [7], which can be extended for detecting most device conflicts in smart cities.

Assuming there are  $n$  services running in a City, an action of a Service  $S_i$  is denoted as  $A_i$ . While a service may have many actions, a general action  $A_i$  is defined as:  
 $\langle SNo, DevNo, Loc, Eff, STime, Dur, Emp, E/O, Pre \rangle$   
 where,

- $SNo$  indicates a unique numerical identifier of the service that issues the action.
- $DevNo$  indicates a unique numerical identifier of the actuator on which the action is supposed to be taken.
- $Loc$  indicates the location of the device or devices.
- $Eff$  is the set of expected effects caused by the action. Effect depends on the functions of the services and could be a change in states/levels/locations, location warnings, messages, etc.
- $STime$  is the estimated start time of the action.
- $Dur$  is the estimated duration of the action.
- $Emp$  gives the importance of this action. Higher  $Emp$  number indicates higher priority.
- $E/O$  indicates the necessity level of the action. When the service sends multiple actions to fulfill a single objective, each of the actions is either essential ( $E$ ) or optional ( $O$ ).
- $Pre$  indicates the pre-conditions of the action.

From the actions proposed by services, the Device Conflict Detection module analyzes the parametrized action requests to detect potential device conflicts. If  $DeNo_x = DeNo_y$  and the actions have temporal overlap, it indicates that Service  $x$  and Service  $y$  have a shared device. Then the rest of the attributes of the actions are compared to detect whether there is any conflict and the type of potential conflict. Such analysis can be performed using a predefined rule-set or formal methods.

2) *Environment Conflict Detection*: Environment conflicts can arise in primary, secondary, or even tertiary level of effects of actions. This poses serious challenges in detecting such conflicts. For instance, (i) Detecting primary effects is often non-trivial as it requires accurate physical models of different environmental phenomena (e.g., air, water, noise, soil, temperature, humidity) and different services of a Smart City. (ii) Detecting secondary or tertiary conflicts is even more challenging, as they rely on the accuracy of detecting primary effects and understanding how primary effects cause secondary effects. Detecting a secondary effect from a primary effect requires substantial amount of domain knowledge, accurate modeling, proper assumptions, etc. (iii) Irrespective of the accuracy of the physical models or the depth of domain knowledge, finding all possible secondary/tertiary effects involves a certain level of error/noise. Hence, the detection and resolution module requires a safety margin while dealing with such conflicts.

A three-layered design for the Environment Conflict Detection module (Figure 2) designed to address the above challenges is proposed. The layers include (i) Physical Model, (ii) Latent Physical Model, and (iii) real-time Feedback Model.

First, the Environment Conflicts Detection module employs various *Physical Model (PM)* to determine if potential secondary conflicts will arise. These physical models are built based on major parameters of environments and an *Effect*

*Library* of services such as, an atmosphere model, noise model, water model, congestion model etc. The *Effect Library* is built and updated by the watchdog to record all potential primary and secondary effects of services.

Second, the Effect Library needs to be maintained and updated by several real-time *Feedback Models (FM)* adaptively. The effects of actions are not fixed in practice, rather they are influenced by many factors, such as seasons, weather, human behaviors, etc. Each FM collects feedback from sensors and examines the accuracy of effects predicted by the PMs.

Finally, because of the complexity of environments and increasing number of services, existing physical models are not enough to detect all environment conflicts. *Latent Physical Models (LPM)* are needed to enhance the physical models. An LPM collects and runs data mining algorithms to (i) detect more secondary effects on environment caused by the actions, (ii) detect latent relations among services, and (iii) build new physical models indicating different aspects of environments and add them to existing PMs.

3) *Human Conflict Detection*: As discussed in Section III, the detection of conflicts involving humans must assess the following three aspects of conflicts.

*Subjective Decision Making*: Often decision makers of a Smart City make contradictory or negatively interacting decisions. Given that the decisions are processed textually, inference based text analysis and rule based conflict detection approaches are necessary.

*Environment on Physiology*: Conflicts in public health/safety can be the result of adversary effects of actions (e.g., pollution), incompatible interventions/message from multiple services, etc. As these conflicts depend on the physiological state of a person they need accurate human physiological models like [8], that also need to be personalized. In case the interventions from one or more services come in a textual format (e.g., sms/warning message/notifications), it will be necessary to automatically detect any contradictory information, using for example natural language processing.

A separate module inside the Human Conflict Detection module is envisioned, which analyzes all textual interventions from a single person and detects whether there are any conflicts among them. At first, this module must detect whether the interventions overlap with each other by inferring the meaning of the intervention and analysing the topics/entities of the text. Then, it must identify potential linguistic features of textual conflicts including, but not limited to, negation, antonym, numerical mismatch, and so on.

*Person vs. Group*: Human conflicts in a Smart City can occur either on a single individual level or on a population level. The size of population may vary from a single family to all city-dwellers. The severity of conflict and safety-critical nature of conflict may also vary. Thus, they often pose a trade-off between scale and severity. The human conflict detection module does not only detect conflicts, but also specifies (i) the scale/size of conflict (i.e., single person, family, neighborhood), (ii) the severity of conflict, and (iii) the safety-critical nature of the conflict, i.e., what happens if it is not resolved.

## B. Design of Resolutions Module

The main purpose of the proposed resolution approach is to resolve the conflicts in such a way so that the *value of services* is optimized. It consists of an Optimized Decision Making (ODM) module and a Preference Learning (PL) module. ODM contains the definition of basic important policies, whereas PL identifies new policies and new preferences by continuously analyzing past resolution decisions.

1) *Optimized Decision Making Module*: Resolution of conflicts takes several arguments, including, but not limited to, Dynamic Priorities (DP), Predetermined Policies (PP), Learned Policies (LP), and Trade-Offs of Decisions (TO). Resolving conflicts is a non-trivial optimization problem. It is denoted as  $Opt(DM)$  and its objective is to choose the best combination of actions, i.e.,

$$Opt(DM) = Opt(\alpha f(DP) + \beta f(PP) + \gamma f(LP) + \theta f(TO))$$

*Dynamic Priorities* take into account the safety aspects of the actions, emphasis and domain of the actions, current state of the city (environment, human factors and other related parameters), context, cost, and time.

*Predetermined Policies* are defined by governments or individuals in some cases. Some predetermined policies involve assigning service priority during designing a service. Predetermined policies also involve resolving static conflicts, i.e., if a conflict between two services is detected during the design phase, policies for them are defined. However, predetermined policies are only a part of the resolution and should be used with caution. In addition, when a new service is added to the system or any environment context changes, predetermined service priorities may change. The predetermined priority can also be overridden by dynamic priorities.

*Learned Policies* are updated by the preference learning module, which learns preferences from users and cities.

*Trade-offs of Decisions* are considered from different aspects. These include comparing resulting costs of different resolution decisions and analyzing the effects of each possible resolution decision. The goal is to maximize service value. This is ensured by running as many services as possible even with some delay, instead of blocking most services.

2) *Preference Learning Module*: The Preference Learning Module learns from preferences of both city-dwellers and city authorities to improve the performance of the Optimized Decision Making module. With humans in the loop, the watchdog resolution module collects users' feedback. Direct feedback is obtained from (i) the collection of city managers' decisions when the watchdog cannot resolve the conflicts and (ii) public or individuals inputs. Indirect feedback is learned from users' operations after conflict resolutions. After a resolution is given from the watchdog, if users disagree with the decision and choose another action, the PL module records this feedback and uses it to improve the parameters of ODM module.

After integrating the decisions from DP, PP, LP and TO,  $Opt(DM)$  needs to take the level of granularity of service actions into account for the final decision [9]. To illustrate the idea, consider an action that requires a group of devices. However, conflict between it and another service only happens

on a part of these devices. In this case, if the actions are analyzed in a fine grained fashion, possibly both actions can be fulfilled immediately or after considering some delay. On the other hand, if the actions are analyzed in a coarse grained fashion, only one action can be fulfilled due to apparent shared resource constraint. Therefore, resolution decisions need to consider conflicts at a high granularity setting.

Finally,  $Opt(DM)$  outputs are one of the following. (i) Accept the higher priority action and reject the other conflicting ones. (ii) Instead of rejecting an action completely, partially perform the lower priority action that is not in conflict. (iii) Send warnings to the services and users to alert them about the conflicts. Although some of the resolutions are automatic, they are not applicable in all conflicting cases, such as when the watchdog system does not contain sufficient information to make an adequate resolution or some safety or health issues are involved whose cost is hard to estimate. In such cases the situation is presented to the city managers.

## V. EVALUATION

The evaluation for conflict detection in Smart City services is conducted in two parts. First, an emulation analysis is performed that uses real data from the city of Aarhus, Denmark. Since coordinated services and their related data are not currently available, eight typical services are hypothesized and correlated with real city data. The main purpose of this emulation of services upon real data is to demonstrate the high potential for conflicts. Since many companies and researchers are investigating integration of services across many different domains, it can be expected that in the near future more conflicts will occur.

The second part of the evaluation hypothesizes 41 future services across 5 domains Table I. A fundamental analysis is conducted that itemizes the conflict frequency among various combinations of these services. The purposes of this analysis are to demonstrate the high degree of potential conflict in the future and to highlight the lessons learned.

### A. Conflict Analysis Using Real City Data

IoT datasets generated from various sensors in the city of Aarhus, Denmark [10] are used for this analysis. The datasets include vehicle traffic, parking, weather, pollution, cultural events and library events for 61 days (in August and September, 2014). In order to analyze the frequency of conflicts happening across services in a Smart City, it is assumed that 8 common services ( $\{S_6, S_9, S_{11}, S_{13}, S_{15}, S_{17}, S_{18}, S_{19}\}$ ) chosen from Table I are running. These services are chosen because they are either existing policies in cities, or smart applications proposed by researchers in previous papers. By emulating these 8 services on the Aarhus datasets, the number of times and condition when each service is triggered are determined. Then an assessment of conflicts is done by detecting two or more overlapping service actuations that are contradictory. As a result, the number of conflicts is obtained. Since there is no real data for integrated services this emulation approach is reasonable because (i) the environment analyzed



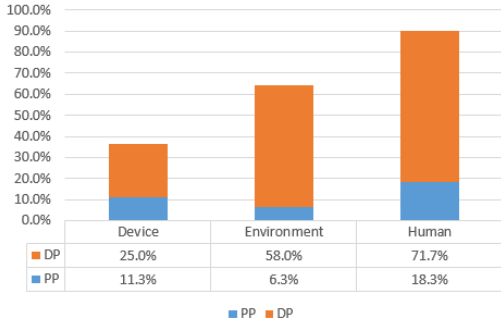


Fig. 3: Conflicts types vs. Resolution.

is from real datasets, and (ii) only typical services are used for emulation.

1) *Conflicts between 2 services*: At first potential conflicts between two services are analyzed. Here, services are assumed to be running separately and the total number of service requests within 61 days is recorded. Then as shown in Table II 9 sets of conflicts were found. For each pair of services in the table, the number of conflicts is shown in column 4. For example, S13 conflicted with S19 32 times. Since these two services are executed a different number of times within 61 days, we also show the percentage of time a service had a conflict with the other service. For example, in row 1 of all the activations of S13, it conflicted with S19 52.5% of its instantiations. Conditions and the corresponding conflicts are also indicated in the table in the last two columns, respectively.

From the results shown in Table II, the following conclusions are drawn:

- Conflicts between 2 services have high frequency, for example, reaching 89% for service 11 conflicting with service 6 (row 9), and being 60.2% on the average for these 8 services.
- The conditions when conflicts occur are very common, but unpredictable.

Furthermore, the average conflict frequency is calculated by combining conflicts from different sources, i.e. device, environment, and human. It is also tested whether a portions of these conflicts could be solved by Predetermined Policies (PP) and Dynamic Priorities (DP) (see Section IV-B). Some conflicts between two services can be solved using the PP approach. For example, the conflicts between smart parking and a building manager service could be solved if the building manager service informs the smart parking service about the events ahead of time. However, many of the conflicts cannot be solved in this way, especially when it concerns environment and human effects.

Figure 3 displays the portion of conflicts that can be resolved using PP and DP approaches for each of the device, environment, and human conflicts. Note that, overall majority of conflicts cannot be resolved using the PP approach. This implies the need for dynamic detection and resolution.

2) *Conflicts among 3 Services*: Conflicts among 3 services are also analyzed. From the conflicts analysis results shown in

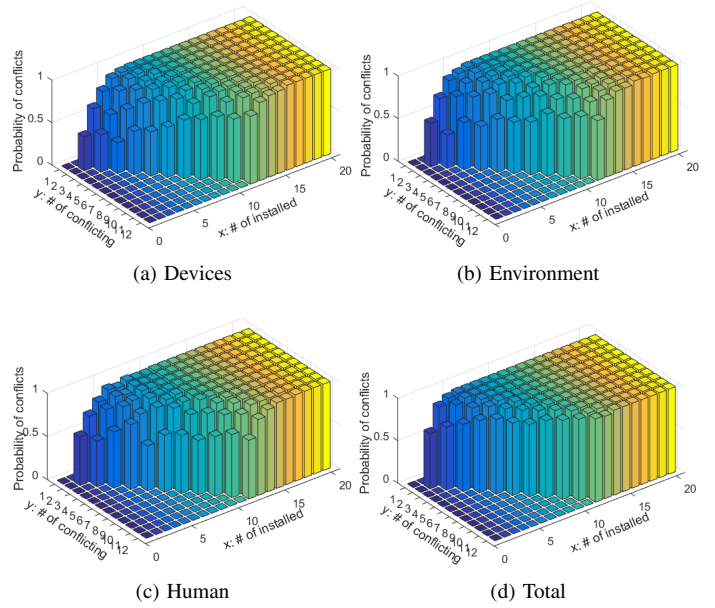


Fig. 4: Probability of conflicts of future services related to device, environment, human, and the total conflicts of all types. x and y axes represent the numbers of installed and conflicting services, respectively.

Table III, the 3-service conflicts also have a high frequency. For example, S11, S17 and S6 conflict almost 70% of the time (calculation is similar to that of Table II). In future smart cities, it is not far-fetched to assume that hundreds of services will be executing concurrently. This data implies that many conflicts are not easily determined ahead of time and must be detected and resolved at run-time.

### B. Conflicts among Future Services

There are 41 future services as presented in Table I. The probability of conflicts among them is analyzed. Because a service can send multiple requests to different devices to complete one task at the same time, there can be overlap among the conflicts.

We randomly select  $n$  services and compute the average probability of conflicts between them. This process is repeated 100 times. Figure 4 shows the probability of conflict between at least  $m$  services when  $n$  services are running from the 41 services in Table II. With the increasing number of services active, the probability of conflicts grows significantly.

Note the starting point (2,2) in Figure 4, i.e., this point represents the probability of conflicts between at least two services when the two services are running. This analysis is performed between two services of each type. The device conflicts have the lowest probability, which is only 31%. By contrast, environment conflicts and human conflicts reach 51%, 78%, respectively. Device conflict is lower as when only two services run simultaneously they are less likely to share a device.



TABLE II: Conflicts Analysis between two Services

	S1	S2	Num. of Conflicts	Conflict Prob. with S1	Conflict Prob. with S2	Condition	Conflict
1	13	19	32	52.50%	63.10%	The weather is bad and big events are going on	S13 discourage vehicles driving in bad weather while S19 sends a large number of taxis to the concert after big events
2	15	17	6	34.60%	47.20%	The parking garage near concert is not available and big events are going on in the concert	S15 adjusts traffic lights to reduce/stop vehicles around concert while S17 directs vehicles to the parking garage of the concert
3	9	17	6	34.60%	47.20%	The parking garage near concert is not available and big events are going on in the concert	S17 directs vehicles to the parking garage of the concert while S9 does not allow any vehicles parking without event tickets during events
4	11	17	1795	70.40%	-	The parking garage is not available and heavy traffic congestion around the nearby parking garage	S17 directs vehicles to the parking garage nearby while S11 tries to solve congestion by adjusting traffic signal
5	6	17	2503	98.12%	-	The parking garage is not available and heavy air-pollution around the nearby parking garage	S17 directs vehicles to the parking garage nearby while S6 limits the number of vehicles around to reduce air pollution by adjusting traffic signal
6	6	18	10	55.50%	85.30%	The air quality is bad and big events are just over	S6 limits the number of vehicles around to reduce air pollution by adjusting traffic signal while S18 sends more buses to the concert station after big events because larger flow
7	13	18	32	52.50%	63.10%	The weather is bad and big events are just over	S13 discourage vehicles driving in bad weather while S18 sends more buses to the concert station after big events because larger flow volume
8	6	19	10	55.50%	85.30%	The air quality is bad and big events are about to be over	S6 limits the number of vehicles around to reduce air pollution by adjusting traffic signal while S19 sends a large number of taxis to the concert after big events
9	11	6	650	75.10%	89.20%	Heavy traffic congestion on street $i$ and air quality is bad on the nearby streets	S11 directs vehicles to alternative path to reduce congestion while S6 limits the number of vehicles on that street to reduce the pollution

TABLE III: Conflicts Analysis among three Services

Services	Conflict Prob.	Condition	Conflict
13,19,18	52.5%	Bad weather after big events	Both buses and taxis are sent to the concert while traffic services limits the number of vehicles on that streets
15,17,9	34.6%	When there is a big event in concert and nearby parking garage is not available	When smart parking sends large number of vehicles to park in rush hour, the streets are blocked off and building does not allow vehicles coming in
11,17,6	69.8%	Parking garage is not available, when streets of nearby garage have heavy air pollution and traffic congestion	When smart parking sends large number of vehicles to park, both traffic services and pollution services are directing vehicles off that streets
19,18,6	55.5%	Heavy air pollution after big events	Both buses and taxis are sent to the concert while pollution services are directing vehicles off that streets
11,13,6	56.1%	Bad weather with heavy air pollution and traffic congestion	Three services have different instructions on the same streets

Another important point is  $(x, 2)$ , the number of services running when at least 2 of them have conflicts. This is 10, 7, 12 and 5 for device, environment, human, and total respectively. This analysis indicates that there is a very high chance that two services have conflicts when as few as 5 services are running in the city. However, there might be hundreds or even thousands of services installed in smart cities in the near future.

The last point in the figure is  $(x, 12)$ , the number of services running when more than 10 of them have conflicts with others with 100% probability. This result is that 15, 16, 16 and 15 services are running for device, environment, human and total, respectively. Unlike the values of  $n$ , values of  $x$  are very close. This demonstrates that regardless of the starting probability, probabilities of all types of conflicts increase rapidly with the increasing number of services. The results also indicate that at

least 15 services will have conflicts when 20 random services are running.

## VI. RELATED WORK

Analysis of runtime dependency and conflicts across CPS systems is a fairly new area of research, and we are unaware of any existing approach for detecting conflicts arising among services of a Smart City. The relevant research includes conflict detection in smart homes and human-in-loop systems, and projected design of smart cities.

### A. Conflict Detection in Smart Homes

DepSys [7] is a dependency aware system for specifying, detecting and resolving conflicts among smart home IoT apps. DepSys can detect conflicts that occur across devices, e.g., multiple apps trying to turn a light on and off at the same time using additional meta data called *effect*, *emphasis*, and *condition*. HomeOS [11] is a PC abstraction to improve manageability and extensibility for smart home apps. It exposes services to home app developers with simple abstractions to access home devices and allows easy incorporation of home devices and applications using common protocols (e.g., Z-Wave and DLNA) and many kinds of devices (e.g., lights, media renderers and door/window sensors). SIFT [12] is another safety-centric programming platform for connected devices in IoT environments. It uses a rule-based policy to verify whether conflicts or policy violations can occur within or between apps. These systems obtain a good performance in the detection of device conflicts among services (apps), but only can deal with very simple environment conflicts.

### B. Conflict Detection in Human-in-the-Loop Systems

With the increasing usage of medical and wellness human-in-the-loop apps, it is likely that multiple medical interventions from multiple apps may be conflicting with each other or

posing hazard to human health when applied together. EyePhy [13] is a system for dependency and conflict analysis across human-in-the-loop apps. EyePhy uses a human body simulator named HumMod [8] to model the effect of various medical interventions on the entire human body using more than 7800 physiological parameters. HumMod could also be used to detect the some effects of services on human body in smart cities, but more parameters and mechanisms are needed to detect the conflicts on humans.

### C. Smart City Systems

The main body of literature on *Smart Cities* systems can be categorized into stakeholder analysis [4], [14], [15], [16], case studies of system implementations [17], [18], and system architecture and services [19], [1], [20], [21], [22]. [14] studies the intersection of services from agriculture to traffic management and how they might be institutionally managed. Authors in [17] take a closer at the transformation of Barcelona into a Smart City by evaluating the benefit of the employed services, challenges, and rounds of proposed improvements. The authors of [18] take a broader look at what Smart City services and systems are employed across Europe, what makes the challenges unique, and which ones have found success and the reason of their success.

## VII. CONCLUSION

Conflicting services pose serious safety threats and operational failure in a Smart City environment. This paper focuses on formulating the problem of conflicts. Specifically, it (i) outlines several characteristics of services that contribute towards conflicts, (ii) proposes a conflict taxonomy in terms of origin of conflict, (iii) lists 41 potential services across five domains (transportation, safety, environment, emergency and energy) for a Smart City, and (iv) outlines issues and research challenges of detection and resolution of conflicts.

In addition, a watchdog architecture is designed for intercepting actions from all services and detecting and resolving conflicts. The conflict detection module contains designated modules to detect device, environment, and human conflicts. Our evaluations using real data and 41 hypothesized services demonstrate the high probability of conflicts in smart cities of the future.

## ACKNOWLEDGMENT

This work was funded, in part, by NSF under grants CNS-1527563 and CNS-1319302.

## REFERENCES

- [1] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 1, pp. 81–93, 2014.
- [2] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis *et al.*, "Smart-santander: Iot experimentation over a smart city testbed," *Computer Networks*, vol. 61, pp. 217–238, 2014.
- [3] A. Bask, K. Spens, G. Stefansson, and K. Lumsden, "Performance issues of smart transportation management systems," *International Journal of Productivity and Performance Management*, vol. 58, no. 1, pp. 55–70, 2008.
- [4] S. Zygiaris, "Smart city reference model: Assisting planners to conceptualize the building of smart city innovation ecosystems," *Journal of the Knowledge Economy*, vol. 4, no. 2, pp. 217–231, 2013.
- [5] J. Chinrungrueng, U. Sunantachaikul, and S. Triamlumlerd, "Smart parking: An application of optical wireless sensor network," in *SAINT Workshops 2007. International Symposium on Applications and the Internet Workshops, 2007.* IEEE, 2007, pp. 66–66.
- [6] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, "Smart cities of the future," *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 481–518, 2012. [Online]. Available: <http://dx.doi.org/10.1140/epjst/e2012-01703-3>
- [7] S. Munir and J. A. Stankovic, "Depsys: Dependency aware integration of cyber-physical systems for smart homes," in *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCCPS)*. IEEE, 2014, pp. 127–138.
- [8] R. Hester, A. Brown, L. Husband, R. Iliescu, W. A. Pruett, R. L. Summers, and T. Coleman, "Hummod: A modeling environment for the simulation of integrative human physiology," *Frontiers in Physiology*, vol. 2, no. 12, 2011.
- [9] P. A. Vicaire, Z. Xie, E. Hoque, and J. A. Stankovic, "Physicalnet: A generic framework for managing and programming across pervasive computing networks," in *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE*. IEEE, 2010, pp. 269–278.
- [10] M. I. Ali, F. Gao, and A. Mileo, "Citybench: A configurable benchmark to evaluate rsp engines using smart city datasets," in *In proceedings of ISWC 2015 - 14th International Semantic Web Conference*. Bethlehem, PA, USA: W3C, 2015, pp. 374–389.
- [11] C. Dixon, R. Mahajan, S. Agarwal, A. Brush, B. Lee, S. Saroiu, and P. Bahl, "An operating system for the home," in *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, 2012, pp. 337–352.
- [12] C.-J. M. Liang, B. F. Karlsson, N. D. Lane, F. Zhao, J. Zhang, Z. Pan, Z. Li, and Y. Yu, "Sift: building an internet of safe things," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 2015, pp. 298–309.
- [13] S. Munir, M. Y. Ahmed, and J. A. Stankovic, "Eyephy: Detecting dependencies in cyber-physical system apps due to human-in-the-loop," in *12th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '15)*, 2015.
- [14] C. Harrison and I. A. Donnelly, "A theory of smart cities," in *Proceedings of the 55th Annual Meeting of the ISSS-2011, Hull, UK*, vol. 55, no. 1, 2011.
- [15] H. Chourabi, T. Nam, S. Walker, J. R. Gil-Garcia, S. Mellouli, K. Nahon, T. A. Pardo, and H. J. Scholl, "Understanding smart cities: An integrative framework," in *2012 45th Hawaii International Conference on System Science (HICSS)*. IEEE, 2012, pp. 2289–2297.
- [16] G. C. Lazaroiu and M. Roscia, "Definition methodology for the smart cities model," *Energy*, vol. 47, no. 1, pp. 326 – 332, 2012, asia-Pacific Forum on Renewable Energy 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0360544212007062>
- [17] T. Bakıcı, E. Almirall, and J. Wareham, "A smart city initiative: the case of barcelona," *Journal of the Knowledge Economy*, vol. 4, no. 2, pp. 135–148, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s13132-012-0084-9>
- [18] A. Caragliu, C. Del Bo, and P. Nijkamp, "Smart cities in europe," *Journal of urban technology*, vol. 18, no. 2, pp. 65–82, 2011.
- [19] I. Vilajosana, J. Llosa, B. Martinez, M. Domingo-Prieto, A. Angles, and X. Vilajosana, "Bootstrapping smart cities through a self-sustainable model based on big data flows," *Communications Magazine, IEEE*, vol. 51, no. 6, pp. 128–134, 2013.
- [20] P. Fritz, M. Kehoe, and J. Kwan, "Ibm smarter city solutions on cloud," *White Paper*, 2012.
- [21] J. Hogan, J. Meegan, R. Parmar, V. Narayan, and R. Schloss, "Using standards to enable the transformation to smarter cities," *IBM Journal of Research and Development*, vol. 55, no. 1.2, pp. 4–1, 2011.
- [22] A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. Koutsopoulos, and C. Moran, "Ibm infosphere streams for scalable, real-time, intelligent transportation services," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 1093–1104.