

Demo Abstract: Simulating Conflict Detection in Heterogeneous Services of a Smart City

Meiyi Ma
Department of Computer Science
University of Virginia
Charlottesville, Virginia 22903
meiyi@virginia.edu

Sarah Masud Preum
Department of Computer Science
University of Virginia
Charlottesville, Virginia 22903
preum@virginia.edu

John A. Stankovic
Department of Computer Science
University of Virginia
Charlottesville, Virginia 22903
stankovic@virginia.edu

ABSTRACT

Despite the increasing intelligence of smart services and sophistication of IoT platforms, the safety issues in smart cities are not addressed adequately, especially the safety issues arising from the integration of smart services. Therefore, in this demo abstract, we present CityGuard, a safety-aware watchdog architecture to detect conflicts among actions of heterogeneous services considering both safety and performance requirements. This demo simulates parts of New York City to depict how CityGuard identifies unsafe actions and thus helps to prevent the city from safety hazards, detects two major types of conflicts, i.e., device and environmental conflicts, and improves the overall city performance in terms of multiple performance metrics. This demo complements the full paper on CityGuard that appears in this conference [2].

CCS CONCEPTS

•Computing methodologies → Modeling and simulation; • Computer systems organization → Embedded and cyber-physical systems;

KEYWORDS

City Safety, Conflict Detection, City Simulation, Smart City

ACM Reference format:

Meiyi Ma, Sarah Masud Preum, and John A. Stankovic. 2016. Demo Abstract: Simulating Conflict Detection in Heterogeneous Services of a Smart City. In *Proceedings of The 2th ACM/IEEE International Conference on Internet-of-Things Design and Implementation, Pittsburgh, PA, USA, April 18-21, 2017 (IoTDI 2017)*, 2 pages.

DOI: <http://dx.doi.org/10.1145/3054977.3057290>

1 INTRODUCTION

With the prevalence of smart cities, increasing number of smart services have been developed independently in different domains, such as transportation, environment, energy, and emergency management. Meanwhile, smart city IoT platforms are built to integrate smart services working under the same system, making the communication and data sharing among them possible and convenient.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IoTDI 2017, Pittsburgh, PA, USA

© 2017 ACM. 978-1-4503-4966-6/17/04...\$15.00

DOI: <http://dx.doi.org/10.1145/3054977.3057290>

However, most of the existing smart services or platforms do not consider safety control in the context of integrated smart services, especially concerning conflicts among them. Not enough attention is paid to the potential safety violations caused by conflicting actions taken by different smart services. For example, while the smart traffic service reduces traffic congestion, its secondary effects might block pedestrians, cause high emission of noise and air pollution to the neighborhood. As a result, unsafe situations exist for the environment and citizens. Safety maintenance in a smart city is extremely important and challenging.

Therefore, CityGuard, a safety-aware watchdog architecture, is created [2] to detect conflicts from multiple smart services in a smart city. It specifies a set of safety requirements and identifies a broad spectrum of conflicts. By running simulations to predict potential conflicts within a specific spatio-temporal range, it detects device and environmental conflicts.

To complement the full paper on CityGuard, this paper is to demonstrate CityGuard by simulating conflict detection in heterogeneous services of a smart city. The demo is presented in the following three scenarios: (i) city without smart services, (ii) city with smart services, and (iii) city with smart services and CityGuard. The results from the simulated smart city scenarios show that (i) seriously unsafe situations and conflicts are caused by secondary effects of the actions of the smart city services, (ii) CityGuard identifies these unsafe conflicts before their occurrences and (iii) improves operational efficiency and safety of the city.

2 CITYGUARD

CityGuard executes in a smart city as a feedback loop between the smart services and the city environment as shown in the Fig. 1. Actions A_1 through A_n from services S_1 through S_n are intercepted by CityGuard. Using the city environment states variables V_1 through V_m , CityGuard runs simulation to predict the effects of an action, A_i . Then it decides if there is a device, and/or environment, or both conflicts based on the safety and performance requirements. Unsafe actions and conflicts are detected through CityGuard. Safe actions are allowed to be performed in the city and cause the change of city states, which triggers actions from smart services again and thus resumes the loop. The four primary components of CityGuard are presented below.

The *Safety and Performance Requirements Component* provides the principles and rules for CityGuard to monitor all the actions, which are integrated into the safety checking components. All principles and requirements are defined and specified by city personnel and can be updated when new services are added and/or new situations emerge in the city.

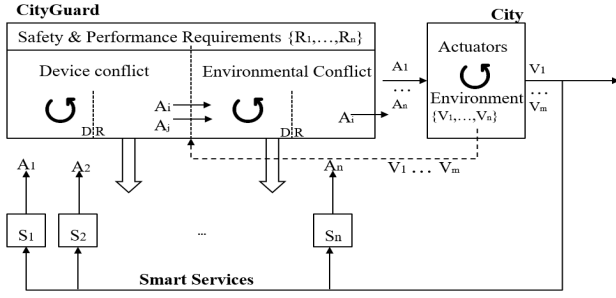


Figure 1: CityGuard Running in a Smart City

The *Real City State and Service Action Component* is the storage component of CityGuard that captures real states and actions of the city. Then it passes them to the Conflict Detection Component for simulation. By this way, consistent view of states in the real city and the simulated city is guaranteed. Also, conflicts of actions are detected in the context of real city states.

The *Pre-processing Component* intercepts actions and obtains their required information including device number, duration, act, and pre-condition. Its purpose is to detect the single environmental conflict. By simulating the actions with the real city states independently, it obtains their potential effects on the environment. Safe actions are sent to the conflict detection components to detect the conflicts among them.

The *Conflict Detection Component* is the central component to detect conflicts and safety violations by simulating the effects of actions on a real city. The potential conflicts of smart cities are categorized as device and environmental conflicts. When more than one action is taken on the same device simultaneously, if these actions are inconsistent with each other, they have a *device conflict*. When one or more actions cause unsafe or contrasting effects on the environment, they have an *environmental conflict*. Definitions and examples of different types of such conflicts can be found in [2, 3]. The solution utilizes Simulation of Urban MObility (SUMO) [1], a traffic simulation that models inter-modal traffic systems including road vehicles, public transport, and pedestrians. To detect unsafe and conflicting actions, CityGuard SUMO simulates real city scenarios and tests the primary and secondary effects of actions. Please refer to [2] for the details of the approaches for detecting the device and environmental conflicts.

3 DEMONSTRATION

The demonstration uses a real map of a part of New York City and generates simulated scenarios based on real data. There are three parts in the demo as follows.

First, simulated Manhattan is shown to present the performance of normal city without any smart service. It results in higher traffic congestion, blocked emergency vehicles, etc.

Secondly, smart services such as the smart traffic service, pedestrian service, air pollution control, noise control and emergency service are implemented in the simulation. In this way, primary and secondary effects of the actions of these services are shown to illustrate their ability to improve the city performance. However, the demo also shows that sometimes they cause unsafe influences

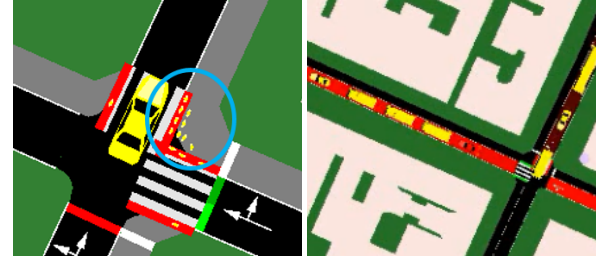


Figure 2: Environmental conflicts. Pedestrians are blocked by the secondary effects of the traffic congestion service in (1). High air pollution is caused by the additive effects of the smart traffic and the smart noise services in (2).



Figure 3: Performances of traffic at the intersection without and with CityGuard. The yellow and red objects denote regular and emergency vehicles, respectively. With CityGuard, there are higher traffic flow, less congestion, and emergency vehicles are prioritized for faster travel time.

on the city when compared to the city without any smart services, as presented in Fig. 2.

The third demo implements CityGuard with all smart services. By comparing the performances of the smart city with and without CityGuard, the demo illustrates that (i) CityGuard detects unsafe situations and conflicts among actions, and (ii) improves the city performance significantly, as presented in Fig. 3. The city performance is measured in terms of multiple performance metrics from transportation and environment domains, such as, the number of vehicles that encounter traffic congestion, the average waiting time of emergency vehicles, and the emissions of noise, CO and $PM_{2.5}$.

ACKNOWLEDGMENTS

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

REFERENCES

- [1] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. 2011. SUMO—simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.
- [2] M Ma, S Masud Preum, and J Stankovic. CityGuard: A Watchdog for Safety-Aware Conflict Detection in Smart Cities. In *2017 ACM/IEEE International Conference on Internet-of-Things Design and Implementation*.
- [3] M Ma, S Masud Preum, W Tärneberg, M Ahmed, M Rutgers, and J Stankovic. 2016. Detection of Runtime Conflicts among Services in Smart Cities. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 1–10.