

# Tackling IoT Ultra Large Scale Systems: Fog Computing in support of Hierarchical Emergent Behaviors

Damian Roca, Rodolfo Milito, Mario Nemirovsky, Mateo Valero

**Abstract** The Internet of Things (IoT) marks a phase transition in the evolution of the Internet, distinguished by a massive connectivity and the interaction with the physical world. The organic evolution of IoT requires the consideration of three dimensions: scale, organization, and context. These dimensions are particularly relevant in Ultra Large Scale Systems (ULSS), of which Autonomous Vehicles is a prime example. Fog Computing is well positioned to support contextual awareness and communication, critical for ULSS. The design and orchestration of ULSS require fresh approaches, new organizing principles. A recent paper proposed HEB (Hierarchical Emergent Behaviors), an architecture that builds on established concepts of emergent behaviors and hierarchical decomposition and organization. HEB's local rules induce emergent behaviors, i.e., useful behaviors not explicitly programmed. In this chapter we take a first step to validate HEB concepts through the study of two basic self-driven car "primitives": exiting a platoon formation, and maneuvering in anticipation of obstacles beyond the range of on-board sensors. Fog nodes provide the critical contextual information required.

## 1 Introduction

The Internet of Things (IoT) represents a phase transition in the evolution of the Internet, characterized by the massive connectivity of endpoint devices (sensors and actuators), and, even more significantly, for the active interaction with the physical

---

Damian Roca\* and Mateo Valero  
Barcelona Supercomputing Center (BSC-CNS) and the Universitat Politècnica de Catalunya (UPC), e-mail: {name}·{surname}@bsc.es

Rodolfo Milito  
Cisco Systems, Inc. e-mail: rodolfo.milito@gmail.com

Mario Nemirovsky  
ICREA Senior Research Professor at BSC-CNS, e-mail: mario.nemirovsky@bsc.es

world [1]. “Things” capture contextual information from their environment and act upon it. Enabling these functionalities poses new requirements to the underlying infrastructure. Given the highly distributed nature of the “things”, they require real time capabilities such as processing and storage close to where the data is generated. It is widely recognized that security is essential for IoT to reach its full potential.

We are at the dawn of IoT, and envision exciting developments. These developments promise huge benefits although they also offer major challenges. While sensors and actuators occupy today our attention, IoT will evolve organically and get transparently involved in most human activities [2]. This organic evolution of IoT requires the consideration of three dimensions: scale, organization, and contextual awareness [3].

These dimensions are particularly relevant in Ultra Large Scale Systems (ULSS) [4], of which Smart Cities [5], and Autonomous Vehicles (terrestrial, aerial, marine, and submarine) [6] are prime examples. Throughout this Chapter we focus on the latter to highlight concepts and architectural foundations. Orchestrating and managing a small set of autonomous vehicles combined with human drivers at modern cities already brings many problems. When that number scales to the majority of circulating vehicles those problems will only aggravate. To ensure a proper operation regime requires handling vast amounts of contextual information for each car to decide its trajectory avoiding collisions.

The stringent latency requirements associated with autonomous vehicles suggests distributed platforms rather than the Cloud for their management [7]. Fog Computing [8] has long recognized the value of extending the Cloud to the edge of the network, bringing networking, compute, and storage resources at different hierarchical levels to respond to the needs of applications and services. Fog addresses the infrastructure and orchestration issues regarding the computational resources [9] (i.e. processing, storage, communications) both at the edge and at different levels of the hierarchy.

A recent paper [10] proposes a fresh approach to design and manage Autonomous Vehicles (AVs) at scale. More specifically, it proposes HEB (Hierarchical Emergent Behaviors), an architecture that builds on established concepts of emergent behavior and hierarchical decomposition and organization [11]. Useful behaviors can emerge from the application of carefully crafted, well understood, and easy to implement local rules. By leveraging emergent behaviors, HEB brings two major benefits. The first, obvious one, is the bypassing of the need to develop highly complex algorithms. The second, perhaps less obvious, but more important benefit is that HEB’s intrinsic flexibility has the ability to handle unanticipated corner cases. The price to pay for these benefits is the need to develop tools (e.g., simulators) to test the emergent behaviors.

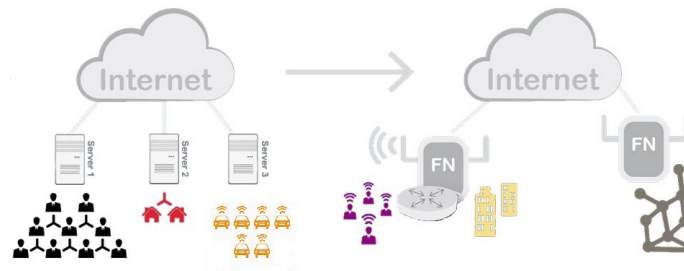
The paper on “Emergent Behaviors in IoT” [10] outlined an agenda to deal with ULSS, with emphasis on AVs. This Chapter advances the agenda in several significant ways: a) developing the concept of “emergent behavior primitives”, and studying the maneuvers of vehicles exiting a platoon and anticipating to obstacles beyond sensors range; b) emphasizing the role of Fog Computing as support for HEB communications in general, and facilitating contextual awareness in particular.

The rest of the chapter is organized as follows. Section 2 refreshes the main architectural foundations of Fog Computing and its advantages. Section 3 revisits and extends HEB. It emphasizes contextual awareness, and introduces the concept of “emergent behavioral primitives [12]”, including the synergies between HEB and Fog Computing. Section 4 is dedicated to the study of simple, but fundamental primitives: maneuvers by vehicles in a platoon to leave the formation exiting the highway and to anticipate and react to obstacles beyond the sensors range. We discuss the promising results, which suggest that the richness and flexibility of the local rules surpass our expectations. Section 5 closes the chapter with the conclusions. Given that this is the first step after the HEB program was announced, we devote some space to discuss open questions and lines of research.

## 2 Fog Computing

Fog Computing is a hierarchically organized architecture of compute, storage, and communication resources that extends from the Cloud to the edge of the network. There are a number of applications and services (e.g. streaming) that can take advantage of Fog. However, from its inception, Fog has been linked to IoT [8, 13].

The true potential of Fog Computing lies in the implementation of a generic multi-tenant platform supporting a wide range of applications simultaneously [14]. Fog breaks down traditional proprietary silos and enables a generic IoT infrastructure, as depicted in Figure 1.

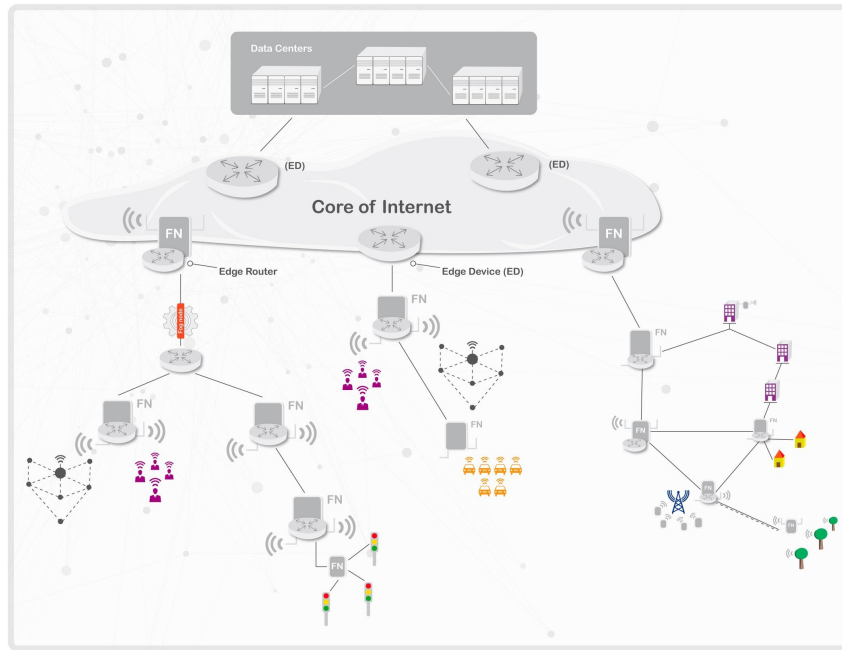


**Fig. 1** Moving from a silo-based implementation to a generic Fog infrastructure capable of supporting multiple applications simultaneously

### 2.1 Fog Computing Architecture

A representative architecture of a generic IoT infrastructure is depicted in Figure 2. At the lower levels are the “things”, sensors and actuators responsible of gathering

information, and acting on the environment. The next layer is formed by heterogeneous Fog nodes, which constitute the aggregation points. The “things” and the nodes communicate mostly through wireless technologies, since both “things” and nodes can move. Due to the Fog nodes’ wide geographic deployment and their location, they can offer resources in real time by processing the data close to where it is generated. The Fog nodes form an interconnected hierarchy. In most cases higher nodes have larger pools of resources at the cost of an increased latency. The Cloud constitutes the highest layer, offering a large pool of resources at low-cost without any latency guarantees.



**Fig. 2** Illustrative example of a generic Fog-based infrastructure serving multiple IoT applications. Fog nodes are interconnected forming a hierarchy.

## 2.2 Fog’s role within ULSS

Having laid out the foundations of the architecture we can explain the role that Fog nodes can play within ULSS. These systems can exploit the locations of the Fog nodes and their hierarchical organization to communicate and to become aware of their context. Thanks to their interconnected architecture, Fog nodes have visibility over a wider geographical range than the one available to individual “things”. For

example, in the AVs use case, a Fog node at the bottom of the hierarchy aggregating data from a set of vehicles has information of a broader area than each individual vehicle. In contrast, each car manages contextual information on a narrower area limited by the range of its sensors and its immediate neighbors in the road. In addition to the broader geographical range, Fog nodes can provide a deeper vision in time. For example, historical information on the traffic conditions.

This visibility places the nodes as notorious information distribution points, fact reinforced by its hierarchical organization (the higher the node, the higher its scope). Then, Fog nodes can transmit information such as the road conditions to optimize cars' trajectories in real time.

Following the reverse process, "things" can use the Fog nodes for their own functionalities. "Things" under a Fog node coverage can use the computational resources of that node to analyze measurements or perform other tasks. When "things" move out of the range they get disconnected from that node. If another node is available in the next location, the same process can continue. This technique eliminates the need of migrating data from one node to the next one because the "thing" itself carries the necessary information.

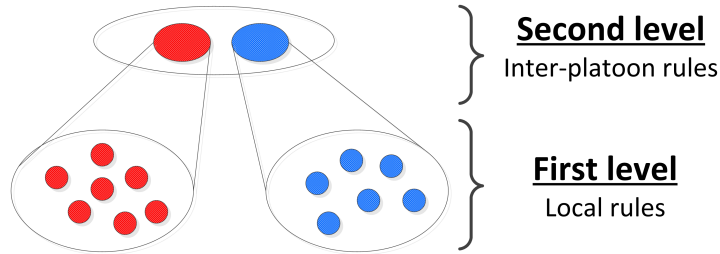
### **3 Hierarchical Emergent Behaviors, a fresh approach for ULSS**

ULSS present architects and developers unique challenges not only because of their massive scale (the number game [4]) but also because of their richness (the diversity of the possible scenarios). The first paper laid HEB's architectural foundations, and its organizational principles. This Chapter offers concrete evidence of HEB's value by exploring in depth basic maneuvers of autonomous vehicles.

#### ***3.1 HEB architecture***

HEB builds up on top of two concepts, namely, emergent behaviors and hierarchical decomposition. The former concept induces behaviors through a set of local rules that define interactions between neighboring "things". The latter organizes complex systems into different levels, each abstracting the essential functionalities of the previous one while maintaining its functionalities. The combination of both areas results in "things" applying lightweight rules that define their interactions with other "things" and also with their environment, as depicted in Figure 3.

Within each rule there is a set of hyper-parameters, e.g. separation distance. Their values can be established at configuration time or dynamically. The main outcome of this technique is the leveraging of the decision process to the "things" themselves. Rather than attempting to explicitly program all potential scenarios in advance (daunting task in the huge state space), the system has the capacity to react effectively to unanticipated situations.



**Fig. 3** Representation of the HEB paradigm with two levels. The first level rules apply to level 1 elements (e.g. AVs) inducing a platoon behavior. The second level applies inter-level rules over the previous level behaviors (e.g. platoons) to enable complex functionalities.

HEB’s potential best manifests itself when the principal actors in the system are mobile (terrestrial, aerial, marine, and submarine vehicles). Within this class of applications we focus on terrestrial AVs. Rather than explicitly programming the vehicles, we develop simple, well understood local rules that regulate the interactions between neighboring vehicles, as well as with the external world.

A set of local rules does not get automatically tagged to a desired behavior. A careful selection process is required. An expressive, realistic simulation platform, coupled with carefully designed experiments, is a promising tool to experiment with rules, their induced behaviors, and the system reaction to unanticipated events.

This new paradigm demands intra and inter-level communications. Rules depend on those interactions to define the “things” behaviors. Without proper communications between “things” and the proper set of sensing capabilities of their environment, behaviors cannot emerge. In the AVs application, these communication capabilities include the ability to exchange information with neighboring cars and roadside units (RSU) and to measure their relative positions and velocities (LIDAR, cameras). In addition, each vehicle is aware of its own location and velocity (GPS, accelerometers). The sensors are already available, the communication protocols (e.g. DSRC [15]) well developed and tested, and given the current state of excitement in the field, the infrastructure will be deployed in the not too distant future.

### 3.2 *HEB, the next phase*

The application of the three original rules from Reynolds [16] to a set of autonomous vehicles results in the formation of a platoon [17] without explicitly program that behavior. However, these rules do not specify the absolute velocity of the group. Platoon absolute velocity is defined as the absolute average velocity of all the vehicles forming the platoon. This velocity is a crucial metric in autonomous vehicles and highly depends on the context, including the quality of the road, weather conditions, vehicle density, maneuvers, and neighboring platoons among others.

The above considerations strongly suggest the need to define the policy not only in terms of local rules. At the end, a policy maps the information state of the system into an admissible set of decisions. For AV HEB, a policy at any given level of the hierarchy includes:

- Local rules pertaining to the hierarchical level.
- The set of hyper-parameters associated to those rules. This set not only includes parameters such as the rules' weights, separation distance, etc. but also velocity applied to each level (i.e. first level refers to average speed of the cars, at the second level is a vector of velocities for each platoon)
- Contextual information. The challenge is to capture in a succinct way the critical information. This requires analysis and careful experimentation. The issue is the required degree of granularity. Contextual awareness includes car density, weather conditions, road conditions, platoon regime, etc.

Architects can define a policy portfolio with well-known emergent behaviors to implement. Given that contextual information is captured in the policies, the selection process becomes a simple, even a trivial one. There are only a few admissible policies for a given informational scenario. Then, the first set of policies to define are the so called "emergent behaviors primitives".

### 3.2.1 HEB primitives

By primitives we understand basic operations required by vehicles within a platoon. Right now we focus on first level behaviors, but the same concept applies to any level within HEB. Vehicle maneuver without collision or handling autonomous cars that want to take an exit in a highway constitute primary examples of a primitive [17]. Simple as they sound, this requires consideration of different aspects and interactions of HEB components:

- Communications between different entities: (i) vehicle to vehicle, (ii) vehicle to RSU, and (iii) distribution of functionalities within the platoon
- Vehicle announcement of its intent
- Non-intersecting exit trajectories whether one or multiple vehicles leave the platoon
- Emergent behaviors at play: current operating rules, their hyper-parameters, and new individual behavior (i.e. exiting the platoon) affecting the emergent behavior.

Taking a closer look to these aspects of the maneuver without collision, we observe the effect of the emergent behaviors through the separation rule between "things" and obstacles. In this case, a single rule provides us the primitive objective if the proper sensing capabilities are satisfied in each moving vehicle.

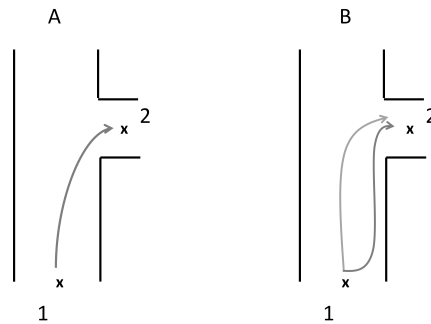
The exiting highway maneuver requires more considerations. Although each vehicle does not know the number of vehicles in the platoon nor has membership awareness, it may notify to its neighboring cars its exit. There is a fundamental

reason for this notification: to avoid an undesired behavior with the entire platoon unconsciously following the exiting vehicle/s.

Communications constitute a key element to ensure a satisfactory maneuver. Efficient exiting strategies necessarily rely on contextual information. It is useful to distinguish between permanent information (coordinates of the exit, proximity to other exits, etc.) from real time information (state of the road, congestion level at the exit, weather conditions, speed of the platoon, vehicular density).

Fog Computing is of great help, from the compute and storage capabilities of the RSU at the edge, to the exchange of real-time information along the RSUs. Fog nodes then become the RSUs in the roads providing their capabilities to the vehicles while building applications on top of their contextual information (i.e. smart guidance systems). Another alternative could be to use the same vehicles to detect and classify the lanes [18].

Last but not least, the use of non-intersecting trajectories in the 3-dimensional space are a must. In traditional solutions with explicitly programmed behaviors, a central orchestrator determines each vehicle's trajectory and makes sure no collision happens. Instead, HEB defines a rule to prevent collisions and gives freedom of choice to the vehicles to decide the best trajectory based on their contextual information. Figure 4 depicts these differences between the two methodologies. In opposition to preset trajectories, HEB approach creates local rules that lead to behaviors emerging in the form of trajectories.



**Fig. 4** Left figure shows an specific trajectory that a vehicle must closely follow. Instead, the right figure shows the freedom HEB leverages to the vehicles to decide their trajectory. Vehicles are capable of taking best decisions since they have all the contextual information.

## 4 Two Autonomous Vehicles Primitives Case Study

AVs constitute one of the clearest HEB applications. The mobility of the vehicles, the constantly changing situations (i.e. road conditions, weather, traffic conditions),



and the number of cars provide a rich set of interactions from which behaviors emerge.

The universe of AV maneuvers is the composition of a vast set of “primitives”. We approach the validation of the HEB architecture by creating and examining in detail a rich “library” of primitives. Each primitive is defined by a goal, and it is self-contained in that it has the ability to reach said goal. We envision the creation of complex scenarios by concatenating primitives. More precisely, we will consider richer goals that are the composition of simpler goals, and achieve them by concatenating primitives. This Section, which is the first step in this direction, focuses on two primitives that rely on Fog nodes deployed as RSUs.

While a primitive is defined by a goal, its full characterization necessitates the specification of the rules that facilitate the achievement of the goal.

## ***4.1 Methodology***

We chose the Processing simulator [19] to perform our analysis of emergent autonomous vehicles. We take as base the set of rules and the environment from the original HEB article where each vehicle determines its trajectory based on four rules that guide its local interactions ( $R_1$  Alignment,  $R_2$  Separation,  $R_3$  Cohesion, and  $R_4$  Destination).

In this first validation effort we limit our attention to first level primitives, that is, primitives that relate to the formation and lifetime of a platoon, and disable the second level rules, which pertain to the interactions between platoons.

Triangles represent the vehicles. Obstacles define the shape the highway with a sideways where the primitives are tested, shown as black dots in the canvas.

## ***4.2 Emergent Autonomous Vehicle Primitives***

The two primitives under evaluation are the exiting maneuver in a highway and anticipating and reacting to obstacles beyond the sensors range. There are many ways of implementing a primitive. In the following Sections we perform a preliminary analysis and present tentative solutions that satisfy the objectives of these two primitives.

### **4.2.1 Exiting maneuver**

In this primitive one or more vehicles in a platoon decide to leave the platoon and exit the highway. We consider the primitive is accomplished satisfactorily if the vehicles exit without collisions or hazardous maneuvers and the rest of the platoon continues the journey unperturbed [20]. Exiting brings forth the interplay between

emergent behavior, classical trajectory design methodologies, and inter/intra layer communications:

- Communications: they were explained before in detail, mainly focused on V2V and V2I [21] enhanced with each vehicle’s sensing capabilities.
- Emergent behaviors: Platoon behavior (induced through the three original Reynold’s rules) and moving objects start roughly at the same speed along the road. As the exiting maneuver proceeds, the velocity vector of the moving object changes in direction and magnitude, but not in a brusque way.
- Trajectory design: Vehicles in the platoon as well as exiting vehicles are interacting but autonomous decision makers, in that they sense the environment and react accordingly. The strategy of fixing exiting trajectories and relying on the collision avoidance ability of vehicles in the platoon seems sound and straightforward. This strategy regards only the platoon vehicles as interacting autonomous decision makers, as depicted in Figure 4.

Among the possible implementations of the exiting primitive we analyze three possibilities of different complexity and observe their impact on the behaviors of interest. The first implementation starts with vehicles notifying their intent to leave the platoon. Since there is no central control of the platoon or membership awareness each vehicle needs to handle the notification to its surrounding vehicles. An intuitive way of notifying its intention is to change the vehicle’s role within the platoon. Instead of being perceived as a vehicle, and therefore subjected to the three platoon rules ( $R_1$ ,  $R_2$ , and  $R_3$ ), perception changes to that of a moving obstacle. In this case, the rest of the vehicles within the platoon avoid it by simply following the non-collision rule ( $R_2$ ). This technique results in exiting vehicles creating a virtual path within the platoon till they make their exit. The vehicle’s new role allows it to leave the platoon and take the desired exit without compromising the platoon behavior for the remaining vehicles.

The challenge is to effect that change of role (from a vehicle in the platoon to moving object) without affecting the local rules or resorting to a central orchestrator. Visualize the scenario in which a platoon of vehicles moves along a highway as depicted in Figure 5. A RSU notifies the platoon of the existence of an exit ahead. The RSU is actually a Fog node, part of a full Fog hierarchy deployed along the highway. The Fog node keeps contextual information, including obstacles in the road ahead, congestion levels, weather conditions in the area, etc. as part of the rich information exchanged with other Fog nodes, both at the same and higher hierarchical levels. Hence, the Fog can extend a vehicle “vision” beyond the capabilities of the on-board sensors.

A vehicle decides to take the forthcoming exit, and broadcasts its neighbors the change of its role, from a peer in the platoon, to mobile obstacle. It does so through the V2V communication channel (e.g., DSRC). From that moment on, that vehicle is perceived as an obstacle by any vehicle happens to be in its neighborhood. As the exiting vehicle maneuvers, its neighborhood changes, but as the notification of its role keeps active, the new neighbors keep away from it. Hence, the exiting vehicle carves a wormhole through the platoon that leads to the exit.



**Fig. 5** Scenario to evaluate the exiting maneuver. It consists of a highway with an exit. The autonomous vehicles can either exit or continue in the highway based on their final destination. The RSU deployed as a Fog node assists with the contextual information.

The same methodology applies when more than one vehicle wants to take the exit. In this case, each exiting vehicle acts individually and it is not coordinated with the other exiting entities. We exploit the power of the rules and their flexibility. Since each vehicle avoids obstacles ( $R_2$ ), there will be no collision among vehicles whether they are part of the platoon or they are leaving. There is no need to implement costly orchestration mechanisms to anticipate all possible situations in micro detail, we just give basic rules and let the vehicles decide what is best for them. The result is a set of vehicles “leaving” the platoon and taking the exit, while the rest of the platoon moves along the highway to its destination.

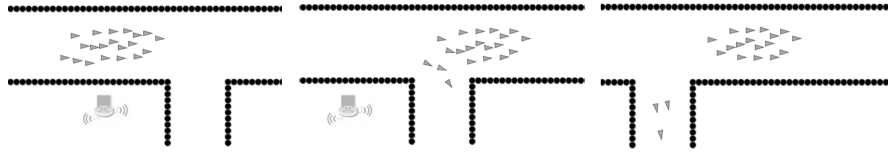
The second implementation uses on a more direct approach based on the rules and their hyper-parameters. This solution does not require extra communications (i.e. notifications) to achieve the primitive’s objective. When the RSU announces the exit that one or more vehicles want to take, the exiting vehicles modify their destination target in  $R_4$  and simultaneously modify the weight associated to that rule. Recall that weights define priorities among the rules that determine the local behavior. The separation rule ( $R_2$ ) still keeps the highest priority to ensure no collision happens but the destination rule dominates ( $R_4$ ) over the rest ( $R_1$  and  $R_3$ ).

While this approach also produces the desired result (vehicles exiting the highway without collisions), we observe differences in the behaviors, which may affect the time it takes to exit the highway. Giving priority to the destination rule over the traditional platoon rules ensures that the vehicles take the exit instead of continuing as part of the platoon. Similar groupings based on destination were analyzed by Hall et al. [22]. What changes with respect to the previous case is how the local rules apply. While the previous technique is based on vehicle to object interactions, the second one relies on rules between vehicles.

We experimented with a third approach, which is in fact a particular case of the previous one: exiting vehicles modify their destination targets, but they do not alter the weight of the rule. The fact that this approach meets the goals of the exiting primitive highlights the surprising expressiveness of the local rules. Adding a target destination rule ( $R_4$ ) we can induce many useful behaviors. Besides the obvious be-

havior of reaching a destination, vehicles with different targets can form a common platoon and later one split to reach both destinations.

Figure 6 depicts a temporal representation of the exiting maneuver implementing the third technique (the results from the previous two are the same except the aforementioned differences). The left figure shows the platoon at the beginning of the highway just before crossing the RSU that communicates the forthcoming exit. The center figure shows some vehicles “leaving” the platoon. Finally, the right figure shows the small platoon of exiting vehicles as well as the platoon of remaining vehicles in the highway. Simulation details not in the figure show exiting cars moving to the edge of the platoon, positioning themselves for a smooth exit, without vehicles crossing their paths. This behavior, not explicitly programmed, emerged naturally from the local rules. It is actually the result of some rules dominating others (in this case  $R_4$ ).



**Fig. 6** Sequential representation (from left to right) of a platoon executing the exiting primitive assisted by the RSU.

The policy emerging with the third technique has considerable degrees of freedom. Consider for instance the rare case in which an exiting vehicle finds the exit suddenly blocked by vehicles ahead. The vehicle cannot force its way out because it is not acting as a mobile obstacle, but rather as a peer of the other vehicles. The car can head back, and rejoin the platoon. Note that this would not be the case with the previous techniques, because they are more aggressive.

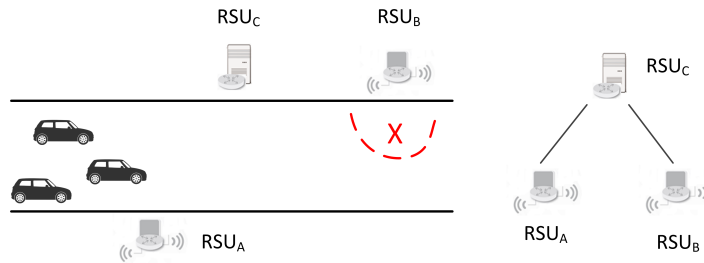
Another remarkable case happens if the platoon does not have a specified target destination at the end side of the highway. In this scenario, when the exiting vehicle executes its maneuver, the rest of the platoon can follow it. This behavior is not problematic though because each vehicle always have a target destination.

A simple set of four rules provides a wide range of useful behaviors that are very robust. In addition these lightweight rules demonstrate an incredible flexibility and simplicity. Fog nodes contribute to handle the contextual information. In this case it only transmits the target destination that will make the vehicles take the proper sideway. We have seen three different ways of implementing this primitive but there are more options that can be part of the policy portfolio (rules, hyper-parameters, contextual information) and can be reused for other applications, reducing the deployment time.

### 4.2.2 Anticipating and reacting to obstacles beyond the sensors range

In this primitive a vehicle or a set of them is circulating in a highway and there is an obstacle beyond the onboard sensors range. The main objective is to anticipate its detection and react accordingly to overcome it without compromising the safety of the driving. Overcoming obstacles brings forth the same areas as the exiting maneuver with the difference that this primitive directly targets the hyper-parameters within the emergent behavior rules.

This scenario is slightly different from the previous ones. We have a platoon moving along a highway as depicted in Figure 7. Along the road there are a set of RSUs that gather information about the road conditions, weather, and traffic among others. In summary, these Fog nodes manage contextual information related to the highway. These nodes are organized hierarchically to capture the information of a wider area. Figure 7 also depicts the virtual architecture they conform with two different levels. The first level is formed by the RSUs closer to the highway, the nodes that physically deal with the vehicles. On the second level there is a single RSU that aggregates the information from the previous level. This level 2 node has a wider scope but its granularity is coarser.



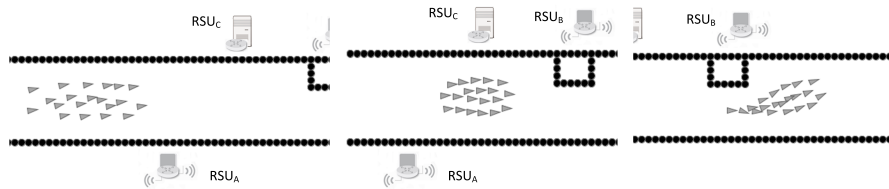
**Fig. 7** Scenario to anticipate and react to obstacles beyond the sensors range. It consists of a highway with a three RSUs organized hierarchically. The right side of the Figure details this architecture with two nodes at the bottom that directly communicate with the “things”, and a higher node to aggregate all the data. The cross represents a temporal blockade in the road.

This hierarchical RSU configuration provides information to the vehicles that traverse the boundaries of locality. While vehicles only sense its closest surroundings, RSUs have a larger scope and can transmit that information to the vehicles. In this situation, vehicles can prepare for the upcoming obstacle or other events. The procedure is as follows:  $RSU_B$  senses the blockade and besides notifying it to the vehicles within its range, it sends this information to the higher layer node,  $RSU_C$ . Then, this node can transmit the information to the other lower level RSUs to take proper actions. In this scenario,  $RSU_C$  sends the notification to  $RSU_A$  that is the node at the left. Now,  $RSU_A$  has the information on the road status ahead and can notify it to the nearby platoon.

To optimize the reaction of the platoon to the blockade,  $RSU_A$  acts upon the hyper-parameters modifying the separation distance between the vehicles and also

their speed. This action influences the emergent behavior in real time. We need to be careful not to augment this distance over the sensing capabilities of each vehicle, fact that will preclude the formation of the platoon. On the other hand, too small a value could result in collisions. Other factors such as the number of lanes in the road also place constraints over this hyper-parameter. In this analysis we keep this distance between acceptable boundaries that do not compromise the behavior. Reducing the distance we induce a compact platoon that can overcome the obstacle easily. To induce proper trajectories,  $RSU_A$  establishes a destination point through  $R_4$  to overcome the blockade smoothly. We are influencing the behaviors through its rules with the final objective of reducing the reaction time.

Figure 8 presents a temporal sequence on how the cars execute this primitive. The left figure presents the initial position of the platoon moving the highway in normal operation regime mode. The center figure shows the modified behavior after  $RSU_A$  has modified the separation hyper-parameter, the speed, and the destination target of the platoon. As you can observe, the platoon now is more compact and the vehicles move closer between them. The right figure shows the platoon overcoming the blockade previously notified to it. Finally, once the platoon totally surpassed the blockade  $RSU_B$  reestablishes the original separation distance, speed, and triggers the original target destination.



**Fig. 8** Sequential representation (from left to right) of a platoon executing an anticipated reaction to obstacles beyond the sensors range

The advantages of such a primitive includes a reduction of the time to overcome situations such as partial road blockades and to show how behaviors can be influenced by the contextual information. In this specific situation we achieve it through a hierarchy of RSUs although there are other solutions. The reaction time reduction comes from how vehicles face the blockade beyond their sensors range. If they are not prepared, a part of the platoon uses the blocked lanes in front of them. Once they sense that obstacle they will change their trajectory to avoid it, but they may have to wait till the vehicles on the clear lanes pass through it. The other possibility is even slower, when both vehicles intersect and the absolute velocity drastically diminishes. Instead, if the platoon is more compact and there are fewer cars on the blocked lanes the reaction time is smaller.

### 4.2.3 Concatenation of primitives to express complex behaviors

Judiciously chosen local rules, though simple to understand and implement, have the amazing capability of inducing behaviors not explicitly enunciated. Local rules are also flexible and expressive, enabling the creation of primitives through minor tweaking and additions, as shown in the previous Section. We observe that the exiting maneuver and obstacle anticipation primitives presented are built on top of a proto-primitive, the platoon formation. This observation suggests that complex behaviors can be achieved by chaining primitives.

Here is an outline of the envisioned methodology, to be developed in detail:

- Create a library of primitives.
- Concatenate primitives to design complex behaviors.
- Test the results through extensive simulations using realistic simulation tools.

The fairly extensive literature on self-driving vehicles that follow prescriptive designs [23] methodology can be leveraged in two ways: a) it suggests a list of primitives and complex behaviors to consider; b) the use cases can be used as baselines to compare with the HEB methodology.

## 5 Conclusions

In this Chapter we validated the HEB concepts introduced in [10] with two specific use cases in the AVs domain. We advanced further the HEB approach by introducing the concept of “primitives”, and their concatenation.

The use cases discussed highlight the role of Fog Computing in supporting the vehicle to RSU communication, as well and providing essential contextual information that extends beyond the immediate neighborhood. This Chapter contributions illuminate the fact that HEB remains in a nascent stage, and that rich lines of research remain open:

- The primitive led design concept reinforces the need of work on rich, realistic simulators, able to incorporate real world scenarios and data.
- The concatenation of primitives deserves attention, both at a rigorous development of the methodology, and the practical building of a portfolio of complex behaviors.
- Machine learning promises to play a major role in the tuning of the hyperparameters at the core of the local rules.
- The domain of autonomous vehicles extends well beyond terrestrial vehicle. Aerial, marine, and submarine autonomous vehicles own idiosyncrasies are worth exploring with the HEB approach.

**Acknowledgements** Damian Roca work was supported by a Doctoral Scholarship provided by Fundación La Caixa. This work has been supported by the Spanish Government (Severo

Ochoa grants SEV2015-0493) and by the Spanish Ministry of Science and Innovation (contracts TIN2015-65316-P).

## References

1. L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, 2010.
2. Rodolfo milito, short video in the september 2016 issue of computing now. [Online]. Available: <https://www.computer.org/web/computingnow/archive/iot-data-and-context-discovery-september-2016>
3. G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *International Symposium on Handheld and Ubiquitous Computing*. Springer, 1999, pp. 304–307.
4. L. Northrop, P. Feiler, R. P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivan *et al.*, "Ultra-large-scale systems: The software challenge of the future," DTIC Document, Tech. Rep., 2006.
5. G. P. Hancke, G. P. Hancke Jr *et al.*, "The role of advanced sensing in smart cities," *Sensors*, vol. 13, no. 1, pp. 393–425, 2012.
6. M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *2014 IEEE World Forum on Internet of Things*,.
7. K. Sasaki, N. Suzuki, S. Makido, and A. Nakao, "Vehicle control system coordinated between cloud and mobile edge computing," in *Society of Instrument and Control Engineers of Japan (SICE), 2016 55th Annual Conference of the*. IEEE, 2016, pp. 1122–1127.
8. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.
9. S. Shin, S. Seo, S. Eom, J. Jung, and K.-H. Lee, "A pub/sub-based fog computing architecture for internet-of-vehicles," in *Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference on*. IEEE, 2016, pp. 90–93.
10. D. Roca, D. Nemirovsky, M. Nemirovsky, R. Milito, and M. Valero, "Emergent behaviors in the internet of things: The ultimate ultra-large-scale system," *IEEE Micro*, vol. 36, no. 6, pp. 36–44, 2016.
11. H. A. Simon, *The architecture of complexity*. Springer, 1991.
12. M. J. Mataric, "Designing emergent behaviors: From local interactions to collective intelligence," in *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, 1993, pp. 432–441.
13. M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero, and M. Nemirovsky, "Key ingredients in an iot recipe: Fog computing, cloud computing, and more fog computing," in *IEEE 19th International Workshop on CAMAD*, 2014.
14. Open fog consortium. [Online]. Available: <https://www.openfogconsortium.org/resources/>
15. J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
16. C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIG-GRAPH computer graphics*. ACM, 1987.
17. P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on automatic control*, vol. 38, no. 2, pp. 195–207, 1993.
18. J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, "Detection and classification of highway lanes using vehicle motion trajectories," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 2, pp. 188–200, 2006.
19. Processing simulation framework. [Online]. Available: <https://processing.org/>
20. A. Hsu, F. Eskafi, S. Sachs, and P. Varaiya, "Design of platoon maneuver protocols for ivhs," *California Partners for Advanced Transit and Highways (PATH)*, 1991.



21. Its vehicle to infrastructure resources. [Online]. Available: <http://www.its.dot.gov/v2i/>
22. R. Hall and C. Chin, "Vehicle sorting for platoon formation: Impacts on highway entry and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5, pp. 405–420, 2005.
23. E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.