Problem solutions 1

Alberto Defendi - alberto.defendi@helsinki.fi

February 14, 2022

1

1.1

Adressability the car has a unique vehicle identification number (VIN)

Connectivity the device has to be wired to exchange data with the external world, for now it's not an IoT. This could be extended by adding a radio module, or by using the connectivity from the user smartphone.

Sensing the car is equipped with many sensors and different knowledge can be extracted from those.

Programmability having a computer on board can makes possible to use a programming language to automate task and deliver information to the passenger on the cockpit and to notify for the insurgence of problems.

Future The car could be turned into IoT by connecting it with a server and providing knowledge to a smartphone app, but this depends upon if the car was designed to be extended and if the manufacturer provides the extension modules to the customer, otherwise DIY solutions provide many communications.

Privacy an hacker could access the location of the car, causing serious repercussions on the car and person safety. Fake errors can be generated, or someone could re-calibrate the car and alter the data values to generate fake alerts and fake evaluations. It would be possible to develop an interface between the car and the smartphone to view the current location and unlock it from the smartphone, in this case the access should be protected with strong access control, to avoid someone to steal the car.

1.2

Addressability the fridge has an unique factory number, also the mac address/serial number of the computer could be used

Connectivity the fridge should be placed in a part of the house with good WiFi connection or an Ethernet cable to work. A fast connection may be required if the camera resolution is high.

Programmability since it has a touchscreen, the computer should have a mobile operative system (e.g.. android) and support the most common high level programming languages.

Sensing the camera can check if the fridge is empty (and if infrared, control the temperature of points in the fridge). The microphone/tablet are good interfaces to the user.

Future The manufacturer could add temperature, humidity sensors to inform about the state of the fridge, a gas sensor (for smell detection), an ice sensor (says if the ice is ready), an IR sensor (eggs detection). But new models shall be released if the manufacturer didn't predisposed extensible modules. Reference:

Privacy an hacker could take control trough the OS of the microphone/camera to spy John's private life, and the data of the products in the fridge could be sold to target John's preferences without his consent. The fridge tablet could be equipped with an app store, where could be possible to download video-conferencing tools, so that John could talk with someone from the fridge. Here there could be fraudulent apps that spied John camera/microphone.

2 Smart Traffic

2.1 Usage

The reality that I chose is related to my observation of the public transport service in the metropolitan area of Helsinki. The methodology I used is trying to reverse engineering the transport network to understand the mechanism, since the documentation is not available. This research will focus on the bus vehicles in the service zones (ABCD), while other vehicles are behind our scope. We use [XHL14] and [Jav+18] as sample. HSL offers a route planner that is accessible trough the website Journey planner and which uses data from OpenStreetMap for the maps and data from their vehicles (bus, ferries, trains) to feed a Journey planner (build upon OpenTripPlanner) to compute the arrival time of the next vehicle, and offer a prediction of the best route to the passenger.

2.2 Benefits

In pre-IoT times, the routes timetable was communicated trough the website or with a schedule printed in the bus stop. However, this model is not flexible enough in situations of traffic or when there are disruptions of services.
The advantages of this IoT network, enables the commuter to find faster routes easily with the aid of the knowledge base extracted from the sensors' data. Having a journey planner makes easier to find the fastest path to reach a destination, while suggesting alternative routes in case of disruption of services or when using multiple routes is required to reach a destination. The interface to the service happens trough the use of a website or a mobile app, where is also possible to purchase tickets for the public transport.

2.3 Architecture

The architectural components are:

- bus receiver
- stop transmitter
- on-board computer (gateway)

From my deductions, the implementation doesn't use the GPS for establishing the position of the bus, which can have connection problems in places where the signal is obstructed by objects, such as tall buildings in the city center and also could require more architectural complexity.
Instead, every route has different points (real bus stops), and a signal is triggered when the bus passes at every point S_n, where n is the number of stops in a route. The bus keeps track of the time spent to reach point S_{i+1} from point S_i, and stores the time to adjust the prediction algorithm, which we will not analyze, but supposing that it's Ford-Fulkerson or Dijkstra's, it could be based on a weighted graph, where the weights is the time needed to reach every stop.

These are the three smart objects under analysis. We use:

Bus state transmitter , which is located in the top part of the bus and is equipped with a small receiver that listens to the nearby stop.

The function is listening to nearby stops to detect when the bus has reached a stop, and send it to the gateway (the driver computer), to update the current position of the bus.

The control unit is similar to the one used in bus stops (a small low-power sensor controller) with the difference that this is powered by the bus engine and constantly exchanges data with the on-board PC.

Power source Bus current.

Bus stop **Function** Bus stops in Helsinki have a small data sender which is usually attached to the stop sign. It is equipped with a small antenna that sends its stop code in broadcast in the nearby area of 10mt, so that the bus can listen to the nearby signal and detect when it has reached a certain stop.

The power source of this device requires to be always on, so it takes the power from a battery and from solar power when possible. The device requires little power, so the battery has to be changed in a yearly basis. When the battery is out of energy, a signal is sent to the bus and then forwarded to the server. The battery should operate between -10°c and 40°c, and resist temperatures until -30 in exceptional situations.

The control unit is a simple micro-controller because it has the simple function of sending the same data. It has also the be cheap and be contained in a safe cover to preserve itself from the external agents (e.g weather, vandalism).

The on-board computer is embedded in the bus, and is managed by the driver. It has two important functional aspects:

- Acts as a gateway between the sensors on the bus and the data from the sensors, such as the stop antenna and the passengers' card reader.
- Keeps a log of the current driver and route.

Usually the driver logs-in with her credentials before to start the turn and indicates the route number before to start the route. The data is constantly interchanged between the computer and the HSL servers.

Connection Since HSL operates in the Helsinki metropolitan area, hence not only in zone A and B where the public WI-FI could be used, but also in zones C and D, where the population density is lower, the most reliable connection would be mobile internet, which is covered

The control unit is a fully-functional low-power, low-energy computer. It has to have enough computational power to handle the keyboard input and enable the real-time transmission of data with the company servers.

The power source is the bus engine, so there are not energy constraints.

References

[XHL14] Li Da Xu, Wu He, and Shancang Li. "Internet of Things in Industries: A Survey". In: *IEEE Transactions on Industrial Informatics* 10.4 (2014), pp. 2233–2243. DOI: 10.1109/TII.2014.2300753.

[Jav+18] Farhana Javed et al. "Internet of Things (IoT) Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review". In: *IEEE Communications Surveys Tutorials* 20.3 (2018), pp. 2062–2100. DOI: 10.1109/COMST.2018.2817685.