Annals of Telecommunications: Special Issue on Belief functions in telecommunication and networks manuscript No.
(will be inserted by the editor)

A high-level application using belief functions for exchanging and managing uncertain events on the road in vehicular ad-hoc networks

Mira Bou Farah[1] · **David Mercier**[1] ·
Éric Lefèvre[1] · **François Delmotte**[1]

Received: date / Accepted: date

Abstract This article introduces a high-level system using belief functions for exchanging and managing imperfect information about events on the road in vehicular ad-hoc networks. The main purpose of this application is to provide the most reliable information for the driver from multiple messages received informing the driver about events on the roads. This system and some variants are tested using a Matlab™ simulator. An implementation with Android smartphones using a Bluetooth technology to exchange the messages is also introduced.

Keywords Vehicular Ad-hoc Networks (VANET) · Events on the road · Imperfect information exchange · Belief functions · Information fusion

1 Introduction

Cars are nowadays by far the most common means of transport. But this popularity leads to safety and environmental problems. Research is conducted in parallel in Europe [11,16,18], in the USA [26] and in Japan [12] to develop Intelligent Transportation Systems (ITS) and to improve safety and driver comfort by in particular looking for the reduction of the number of undesirable events on the roads such as accidents and traffic jams.

Ad-hoc networks are wireless networks capable of organizing themselves without any infrastructure. Instead of communicating through a centralized access point, they are composed of wireless nodes communicating with each other to exchange information. In a mobility context, they are called MANET (Mobile Ad hoc Networks) [5,13,20,1].

[1] Université Lille Nord de France, F-59000 Lille, France
UArtois, LGI2A, F-62400, Béthune, France
Corresponding Author: D. Mercier
Tel.: +33 (0)3 21 63 23 49
E-mail: david.mercier@univ-artois.fr

Vehicular Ad-hoc Networks (VANET) are MANET networks applied to Inter-Vehicle Communication (IVC) where nodes are highly dynamic. Two main modes of communication are known in IVC: Vehicle to Infrastructure (V2I) and Vehicle to Vehicle (V2V). A V2V communication example is given by the VESPA system (Vehicular Event Sharing with a mobile P2P Architecture) [2,6,8] which allows vehicles to share different types of information such as accidents or emergency brakings. It guides in particular drivers to find a parking place or to get traffic information. Another example is given by TrafficView [17] a road data dissemination system using V2V communication, allowing continuous exchange of position and speed information between vehicles to inform drivers about road situation (traffic, weather, quickest way, ...).

One important component of V2V communication concerns the data transmission protocol, and currently no universal standard has emerged [7]. Communications can be realized using DSRC (Dedicated Short-Range Communications), Wi-Fi (ad-hoc), Bluetooth, 3G or 4G cellular networks, SDARS (Satellite Digital Audio Radio Service), etc.

The present work concerns V2V communications, where information is decentralized and vehicles build their own information. It is focused on the high-level aspect of merging and organizing the messages received by a vehicle. To demonstrate the applicability of the proposed approach to real systems, a simple Bluetooth system is also developed and tested on-line. Extensions of this approach with other technologies, such as with Wi-Fi devices for instance, is beyond the scope of this paper.

In vehicular networks, the environment is very dynamic. Events on the road appears and disappears over time. Vehicles receive a large amount of information often uncertain, redundant or outdated. Then to manage the exchanges of such imperfect information between vehicles, a method is proposed in this article using the belief functions theory [21,22], belief functions constituting a rich and flexible framework to handle imprecise and uncertain information [9, 19].

Early work integrating belief functions has already been developed by Cherfaoui et al. in [4]. It has been extended in [3] where the authors propose a first model able to manage different events on the same segment of road, and a simple strategy for sending messages is exposed. Each vehicle can either send new messages created by the driver regarding events on the road or transfer messages received from other vehicles, each vehicle building its own map of the current situation without delivering to other vehicles its results of combination or deductions.

In this article, this previous model is deeply extended and tested using a Matlab™ simulator. A new mechanism to model the aging of a message is considered and compared with the first approach which is based on a discounting process considering that the information contained in a message decreases in reliability over time. The new one interprets the aging of a message as a reinforcement in the fact that the event has disappeared as do all events over time. Two strategies for sending messages are also studied: a first one where all messages are kept in vehicles data bases and all messages are exchanged, and a second approach aiming at saving bandwidth and spacing where only one merged message by event is retained in each vehicle. At last, an implementation of this model is realized using Android smartphones and Bluetooth connections to illustrate a simple concrete case of use and to demonstrate that smartphones can manage belief functions calculations.

This article is organized as follows. Basic concepts on belief functions are recalled in Section 2. A credal model for representing and managing imperfect information about events on the road is exposed in Section 3. It is simulated and tested in Section 4. Then, an example of implementation with Android smartphone application using Bluetooth transmissions is presented in Section 5. Finally, Section 6 concludes this paper and gives prospects.

2 Belief Functions: Basic Concepts

In this article, belief functions are handled with Smets' semantic approach: the Transferable Belief Model (TBM) [22,24], where belief functions are interpreted as weighted opinions with no underlying probabilistic measure.

Two levels are distinguished in the TBM:

- the credal level, where available pieces of information are represented by belief functions, and manipulated;
- the pignistic or decision level, where belief functions are transformed into probability measures when a decision has to be made, and the expected utility is maximized.

2.1 Representing Information

Let us consider $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$, called *frame of discernment*, a finite set of possible answers to a given question Q of interest.

A knowledge regarding the answer to question Q can be quantified by a *basic belief assignment (BBA)*, also called a *mass function*, denoted by m and defined from 2^Ω (the set of all possible subsets of Ω) to $[0,1]$ such that the sum of all the masses is equal to 1:

$$\sum_{A \subseteq \Omega} m(A) = 1 \ . \tag{1}$$

Each mass $m(A)$, with $A \subseteq \Omega$, represents the part of belief allocated to the fact that the true answer to question Q belongs to A. In particular $m(\Omega)$ represents the degree of ignorance of the source which has provided the information m.

Each subset A of Ω such that $m(A) > 0$ is called a *focal element* of m.

A BBA m is said to be *categorical* if it has only one focal element A, which means that $m(A) = 1$. It is then denoted by m_A. In particular m_Ω represents the categorical BBA such that $m(\Omega) = 1$ which corresponds to a state of total ignorance for the agent. BBA m_Ω is also called the *vacuous belief function.*

A mass function m having less than two focal elements including Ω is called a *simple mass function* and verifies:

$$\begin{cases} m(A) = 1 - \omega \\ m(\Omega) = \quad \omega \end{cases} \tag{2}$$

with $A \subset \Omega$ and $\omega \in [0, 1]$. It can be conveniently noted A^ω.

2.2 Manipulating Information

2.2.1 Discounting operation

When receiving a piece of information m, an agent may have some doubts regarding the reliability of the source which has provided this BBA. Such a metaknowledge can be taken into account by using the discounting operation [21, page 252] defined by:

$$\begin{cases} {}^{\alpha}m(A) = (1 - \alpha)\, m(A), \quad \forall A \subset \Omega, \\ {}^{\alpha}m(\Omega) = (1 - \alpha)\, m(\Omega) + \alpha\ , \end{cases} \tag{3}$$

or more simply:

$$ {}^{\alpha}m = (1 - \alpha)\, m(A) + \alpha\, m_\Omega\ , \tag{4}$$

where $\alpha \in [0, 1]$ is called the discount rate.

Coefficient $\beta = (1 - \alpha)$ represents the degree of reliability regarding the information which have been provided [21, 15].

2.2.2 Reinforcement operation

On the contrary, an agent may want to reinforce a BBA which seems to be too cautious in the sense that it is not specific enough. This operation can be realized using the following correction mechanism [14]:

$$ {}^{\nu}m = (1 - \nu)\, m(A) + \nu\, m^{tr}\ , \tag{5}$$

where $\nu \in [0, 1]$ is called the reinforcement rate, and m^{tr} is the mass function m totally reinforced.

Transformation m^{tr} of m corresponds to the mass function expected by the agent when m has to be fully reinforced. Different possible choices are given in [14].

In next Section 3.5 a reinforcement process is introduced with BBA m^{tr} chosen equal to a particular m_{ω_i} with $\omega_i \in \Omega$, which means that according to the reinforcement rate ν the agent has some reason to believe that ω_i is the true value he/she searches.

2.2.3 Conjunctive rule of combination

Two mass functions m_1 and m_2 obtained from distinct and reliable sources, can be combined using the conjunctive rule of combination denoted by $\textcircled{\cap}$ and defined by:

$$m_1 \textcircled{\cap} m_2(A) = m_{1\textcircled{\cap}2}(A) = \sum_{B \cap C = A} m_1(B) \cdot m_2(C) \ , \forall A \subseteq \Omega \ . \qquad (6)$$

With this combination, masses are transferred to focal elements intersections.

2.2.4 Cautious rule of combination

If mass functions m_1 and m_2 are obtained from non-distinct reliable sources, they can be combined by using the cautious rule of combination [10], denoted by $\textcircled{\wedge}$ and defined as follows:

$$m^{\Omega}_{1\textcircled{\wedge}2} = \textcircled{\cap}_{A \subset \Omega} A^{w_1^{\Omega}(A) \wedge w_2^{\Omega}(A)} \ , \qquad (7)$$

where $\wedge$ denotes the minimum operator, and w is the conjunctive weight function [23] defined by:

$$w(A) = \prod_{A \subset B} q(B)^{(-1)^{|B|-|A|+1}} \ , \quad \forall A \subset \Omega \ . \qquad (8)$$

The cautious rule of combination is commutative, associative and idempotent (which means that: $m \textcircled{\wedge} m = m$).

2.3 Making a decision

At this level, the mass function m defined on Ω and representing the available information regarding the answer to question Q, and generally resulting in practice from a fusion process, is transformed into a probability measure with the *pignistic transformation* [25] defined by:

$$BetP^{\Omega}(\{\omega\}) = \sum_{\{A \subseteq \Omega, \omega \in A\}} \frac{m(A)}{|A| \ (1 - m(\emptyset))}, \quad \forall \omega \in \Omega. \qquad (9)$$

3 A credal model for representing and managing imperfect information about events

In this section, messages format and the different parts composing the proposed credal model allowing drivers to have a more reliable view of situations on the road based on exchanged messages are exposed.

Vehicles are assumed equipped with a Global Positioning System (GPS). Locations and dates of events are assumed to be automatically generated as soon as drivers need to send messages regarding events.

3.1 Exchanged messages format

Vehicles exchange messages about events on the road, each message giving information regarding one event.

A message M is represented as a 5-tuple (S, t, d, ℓ, m) described in Table 1.

Table 1 Messages format

Attribute	Description
S	Source which has perceived the event
t	Type of the event
d	Date and Time when S has detected the event
ℓ	Location where S has detected the event
m	Mass function representing the confidence of S regarding the fact that the event is present

Each attribute $x \in \{S, t, d, \ell, m\}$ of a message M will be denoted by $M.x$:

- $M.S$ designates the source of information which have perceived the event. It is not necessarily the source which have transferred the message M.
- $M.t$ indicates the type t of the event reported by message M. Different types of events can be considered such as accident, parking place, traffic-jam, fog blanket, animals on the road, etc.
- $M.d$ points out the date when $M.S$ has created the message to inform about its presence. It is generally not the date at which message M has been received.
- $M.\ell$ indicates the location ℓ of the event reported by message M.
- At last, $M.m$ denotes the mass function m, held by vehicle source S, which is defined on the frame of discernment $\Omega = \{\exists, \nexists\}$ where:
  - $\exists$ stands for "the event, which is of type t, is present at time d at location ℓ."
  - and $\nexists$ means "the event, which is of type t, is not present at time d at location ℓ."

3.2 Sending process

Each vehicle manages an inner database composed of the messages it has either received or created.

All the messages M_i^j concerning an event j are regrouped into a table denoted by M^j.

Each time a vehicle can exchange messages with another vehicle, all the messages present in its database are exchanged.

3.3 Reception process

The reception of a message by a vehicle is realized sequentially by the procedure illustrated in Figure 1.

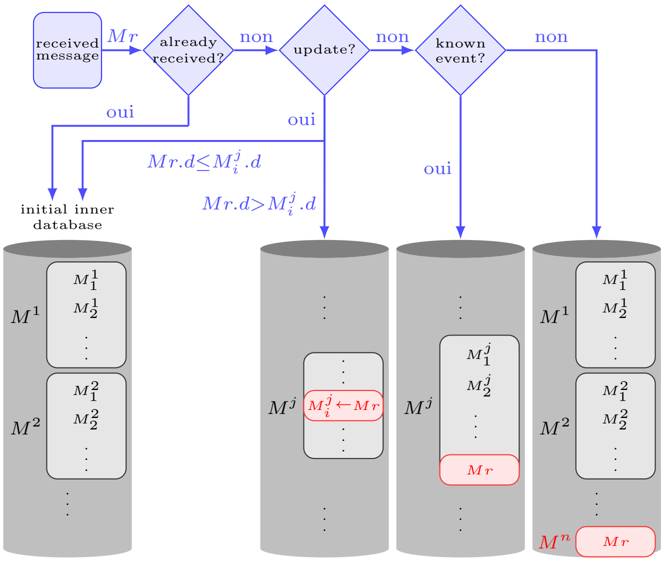


Fig. 1 Messages reception process.

Case of a message already received If the received message has been already received, it is ignored. Otherwise the message goes to the next test.

Case of an update The same source may send a message and then send an update of this message, which means that it sends a new message corresponding to the same event to correct the information previously sent.

Such an update example is illustrated in Figure 2, the date d_2 of the updated message is more recent than d_1, locations ℓ_1 and ℓ_2 of the messages are neighbouring, mass functions m_1 and m_2 may be different or not.

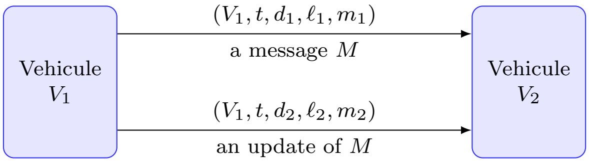


Fig. 2 Illustration of a message update example.

To determine an update, a threshold, denoted λ_t and depending on the type t of the event, is used. It corresponds to the maximum value for a difference of location to consider that a message can be an update of another.

More precisely, the system will consider that a message M_1 is an update of a message M_2 if:

- $M_1.S = M_2.S$;
- and $\mathit{\Delta}(M_1.\ell, M_2.\ell) < \lambda_t$ with $\mathit{\Delta}$ a distance;
- and $M_1.d > M_2.d$.

If a message present in the database is an update of the received message, the latter is ignored. If the received message is an update of a message present in the database, then it replaces it. Otherwise the message goes to the next test.

Grouping messages corresponding to the same event The received message has not been already received and is not an update of a message present in the database, it remains to look for the table M^j in which to put it, in other words to look for which events j it corresponds to.

Let $\mathcal{M}$ be the set of all the messages M_i^j corresponding to an event j of the same type of the event associated with the received message Mr and which is close in location, which means that $\mathcal{M}$ is composed of all the messages M_i^j verifying: $Mr.t = M_i^j.t$ and $\mathit{\Delta}(Mr.\ell, M_i^j.\ell) < \lambda_t$ with $\mathit{\Delta}$ a distance measure.

If $\mathcal{M}$ contains at least one message, the received message Mr is inserted into the same table of the message M_i^j present in $\mathcal{M}$ which is the nearest in location.

If $\mathcal{M}$ is empty, a new event j is created, its table M^j containing only one message which is the received message Mr.

3.4 Messages obsolescence

Over time, all messages stored in tables are not necessarily useful: some of them become obsolete. Then messages having a perception date too old are deleted.

The deletion is realized using a threshold, denoted Del_t depending on the type t of the event.

So, in parallel with the other processes, each message M_i^j such that:

$$\mathit{\Delta}(now, M_i^j.d) > Del_t \ , \tag{10}$$

with $\mathit{\Delta}$ a distance measure, is suppressed.

3.5 Data Fusion: giving an overview of the situation to the driver

From the database composed of the tables M^j, themselves composed of messages M_i^j corresponding to a same event j, an overview of the existing events with their probabilities can be given to the driver, in the following manner:

- First, each mass function $M_i^j.m$ is corrected with a correction rate equal to $\frac{\Delta(now, M_i^j.d)}{Del_t}$ to take into account the aging of the message. The older the message, the more corrected it is.

  Two correction mechanisms can be used:
  - the discounting operation (cf Equation (4)): ${}^{\alpha_i^j}M_i^j.m = (1-\alpha_i^j)\,M_i^j.m + \alpha_i^j\,m_\Omega$, where the correction rate $\alpha_i^j = \frac{\Delta(now, M_i^j.d)}{Del_t}$ corresponds to a discounting rate. With this operation, over time ${}^{\alpha_i^j}M_i^j.m$ tends to the total ignorance m_Ω.
  - the reinforcement process (cf Equation (5)): ${}^{\nu_i^j}M_i^j.m = (1-\nu_i^j)\,M_i^j.m + \nu_i^j\,m_\nexists$, where the correction rate $\nu_i^j = \frac{\Delta(now, M_i^j.d)}{Del_t}$ is a reinforcement rate. With this operation, over time ${}^{\nu_i^j}M_i^j.m$ tends to $m_\nexists$ meaning that event j has disappeared.

  Let us recall that in both cases, messages are discarded as soon as $\Delta(now, M_i^j.d) > Del_t$ (cf Equation (10)).
- Then, for each event j, corrected mass functions (${}^{\alpha_i^j}M_i^j.m$ or ${}^{\nu_i^j}M_i^j.m$) are conjunctively combined using Equation (6).
- At last, pignistic probabilities regarding each event presence are obtained using Equation (9).

These two methods are tested and compared in Section 4.3.1.

An overview of the whole proposed system is given in Figure 3.

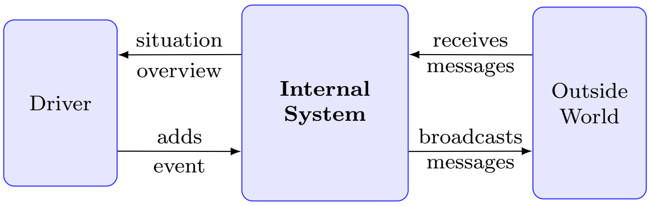


Fig. 3 General flowchart of the model. The internal system manages new messages, repeats messages which have still an interest, deletes useless one, and offers an overview at driver's request.

3.6 Variant: exchanging and keeping only the fusion results in databases

In this section, a variant of the first method is presented to save bandwidth and limit the number of messages stored in databases. In this second model, only the fusion results are kept in databases and exchanged between vehicles.

Only the changes between these two models are exposed in the sequel.

3.6.1 Messages and databases

Each vehicle database contains only one message associated with each event.

A message M^j associated with event j is a 5-tuple (S, t, d, ℓ, m) where:

- $M^j.S$ is the set of sources which have contributed to messages creation concerning the event j;
- $M^j.t$ is the type of the event j;
- $M^j.d$ is the date and time of the most recent message concerning the event j;
- $M^j.\ell$ is the set of locations of the event j;
- $M^j.m$ is the mass function representing the confidence of sources $M^j.S$ regarding the fact that the event is present.

When possible, vehicles exchange their messages, each one being viewed as a synthesis of the information related to an event on the road.

3.6.2 Reception and fusion process

A received message Mr concerning an event j already identified in the vehicle database (they are of the same type and close in location) is fused with message M^j as follows:

- The new set of sources is equal to $M^j.S \cup Mr.S$;
- The date of M^j becomes the most recent date among $M^j.d$ and $Mr.d$;
- The new mass function of M^j is obtained as follows:
  - The mass function of the message having the oldest date among M^j and Mr is corrected to take into consideration its aging (correction rate being equal to $\frac{|\Delta(Mr.d, M^j.d)|}{Del_t}$).
  - If $Mr.S \cap M^j.S = \emptyset$ (before adding $Mr.S$ to $M^j.S$), the new mass function of M^j is obtained from the conjunctive rule of combination Equation (6) of the corrected mass functions of $M^j.m$ and $Mr.m$.
  - Otherwise, the cautious rule, Equation (7), is used to combine these two corrected mass functions.

If the event j is not already identified in the vehicle database, message M^j is created with the attributes of Mr: $M^j.S = \{Mr.S\}$, $M^j.t = Mr.t$, $M^j.d = Mr.d$, $M^j.\ell = Mr.\ell$ and $M^j.m = Mr.m$.

A comparison of the model keeping all the messages and the second model keeping only the fused messages is realized in Section 4.3.2 with both discounting and reinforcement correction operations.

4 Experimental tests using Matlab™ simulations

4.1 Simulator

In order to test and compare the proposed methods, a simulator has been developed in Matlab™ such that different scenarios can be tested.

A scenario is built with the following components:

- A scenario time axis τ with start and end dates, and a sampling period $\Delta\tau$.
- Type, location, start and end dates of each real event on the map.
- Attributes related to vehicles given by:
  - V: the number of vehicles.
  - $NetRange$: the network range in meter allowing V2V communication.
  - $Speed$: the average vehicle speed in km/h when no event is present on the road.
  - $SDRate_t$: the slow-down rate, depending on event type, used to compute new vehicle speed when an event exists at its location at τ moment. In this case, new vehicle speed is equal to $(1 - SDRate_t) \cdot Vit$ in km/h from τ to $\tau + 1$.
  - The following attributes to obtain the path of each vehicle:
    - τ_d: the moment τ of vehicle departure;
    - ℓ_d: the departure vehicle location;
    - ℓ_a: the arrival vehicle location.

At each τ moment, the following actions are realized:

- each vehicle creates a message when an event is perceived;
- each vehicle denies an event present in vehicle database if the driver does not perceive it;
- each vehicle communicates its messages to neighboring vehicles in the network range $NetRange$.

4.2 Performance measure: Adequacy to the reality

The common purpose of the proposed models is to provide the most reliable information for the drivers. Thus performance rates of models are measured for each type t of event and for each vehicle v by the adequacy to the reality of the information given to the driver.

Formally, at each time step τ, the set equal to the union of the events j present in the vehicle database and the existing events in the reality is considered and denoted by $J_t^{v,\tau}$, and performance rates are computed for each type t of event and for each vehicle v at each time τ by:

$$Perf_t^{v,\tau} = 1 - \frac{\sum_{j \in J_t^{v,\tau}} \left(BetP_j^{v,\tau}(\{\exists\}) - R_j^\tau \right)^2}{\mid J_t^{\tau,v} \mid} \ , \tag{11}$$

where:

- $R_j^\tau = 1$ if event j is present at time τ, 0 otherwise.
- $\mid J_t^{v,\tau} \mid$: number of events of type t present in vehicle v database or existing in the reality at time τ.
- $BetP_j^{v,\tau}$: pignistic probability of vehicle v at time τ concerning the presence of the event j. If no message corresponding to event j is present in vehicle v database then $Betp_j^{v,\tau}(\{\exists\}) = 0$.

4.3 Experimentations

The methods exposed in Section 3 are tested and compared through different scenarios.

In all the scenarios exposed here, an accident starts at the beginning of the simulation and lasts 60 minutes, which means that it is present from $\tau = 0$ to $\tau = 900$ ($\Delta\tau = 4$s).

The confidence of all created messages is equal to 0.6: the mass function of the messages created is either $m(\{\exists\}) = 0.6, m(\Omega) = 0.4$ when vehicle confirms an event or $m(\{\nexists\}) = 0.6, m(\Omega) = 0.4$ when vehicle denies an event.

4.3.1 Comparing messages aging mechanisms: discounting and reinforcement

Two scenarios are presented to compare aging mechanisms using the method where all messages are kept in vehicles databases.

Scenario n^{o}1 In the first scenario, a vehicle v receives a message indicating the presence of the accident few minutes after its beginning. After a delay of Del_t equal to 100 minutes, the message is suppressed from the vehicle database being considered as obsolete.

In the following, in each figure illustrating the adequacy to the reality of a vehicle, the messages added to the vehicle database or suppressed from the vehicle database are illustrated by the following symbols just above the time τ where they occur:

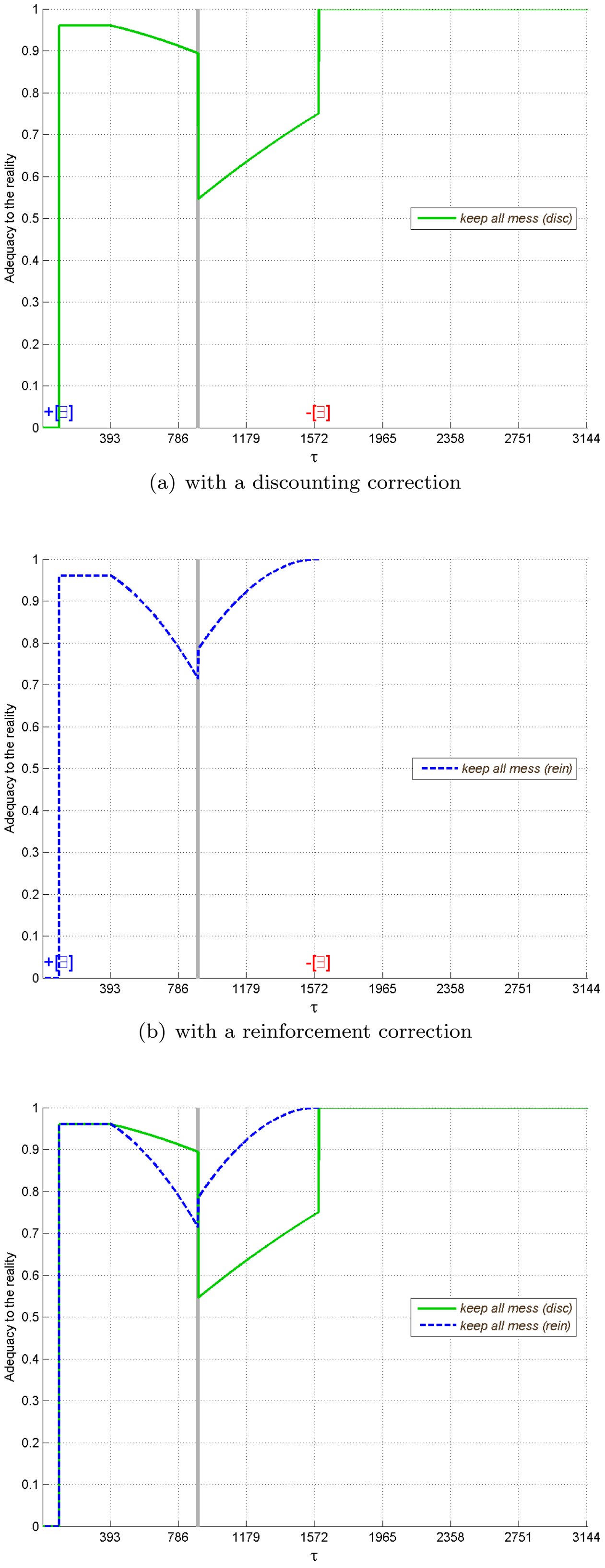
- $+n[\exists]$ which means that n new messages confirming the event are added to vehicle database;
- $+n[\neg\exists]$ which means that n new messages denying the event are added to vehicle database;
- $-n[\exists]$ which means that n messages confirming the event are deleted from vehicle database;
- $-n[\neg\exists]$ which means that n messages denying the event are deleted from vehicle database.

A grey vertical line indicates the disappearance of the event.

Figure 4 illustrates the adequacy to the reality of vehicle v during this first scenario.

The discounting mechanism tends to the total ignorance while the reinforcement process tends to the negation of the event. Therefore, the discounting mechanism gives a better result (a better adequacy to the reality) while the accident has not disappeared (the accident disappearing after the grey vertical line). On the contrary, the reinforcement process which anticipates the disappearance provides better results after the disappearance.

Scenario n^{o}2 In this scenario, a vehicle v receives two messages confirming the accident, and two other messages denying the accident 10 minutes and 30

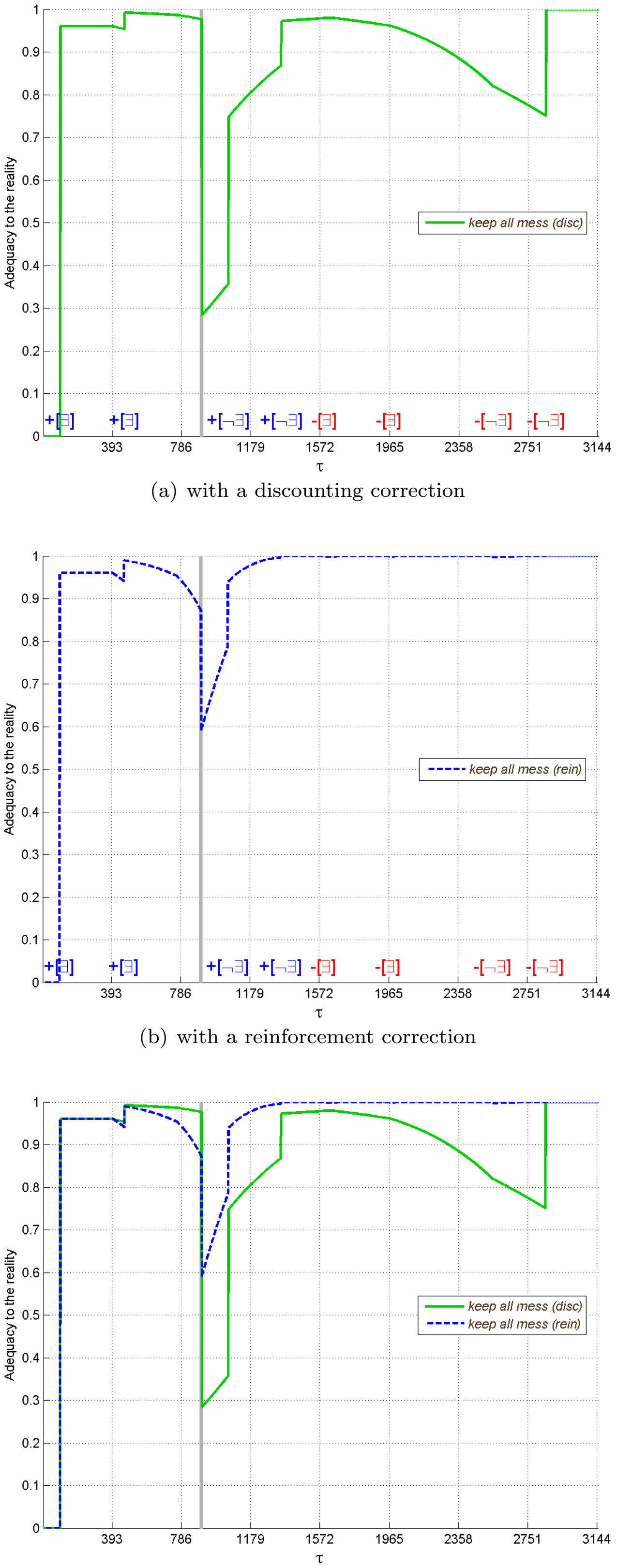


(a) with a discounting correction

(b) with a reinforcement correction

(c) Discounting and reinforcement corrections on the same figure

Fig. 4 Adequacy to the reality of the information given to the driver with the first model (keeping all the messages) during scenario n°1. Symbol $+[\exists]$ indicates the date of the arriving of a message which mentions the accident existence. Symbol $-[\exists]$ indicates the deletion of the message mentioning the existence of the accident. The grey vertical line corresponds to the disappearance of the accident. Accident starts at $\tau = 0$.



(a) with a discounting correction

(b) with a reinforcement correction

(c) Discounting and reinforcement corrections on the same figure

Fig. 5 Adequacy to the reality of the information given to the driver with the first model (keeping all the messages) during scenario n°2.

minutes after its disappearance. The adequacy to the reality of vehicle v is illustrated in Figure 5.

This scenario shows that after receiving messages denying the accident (from $\tau > 1050$, 10 minutes and 30 minutes after the accident disappearance), adequacy tends quickly to 100% with the reinforcement process while it is not the case with the discounting process. Indeed the discounting process tends to transfer the masses to the ignorance giving a 50% bet for the existence and 50% to the non-existence of the event.

4.3.2 Comparing methods: keep all messages, keep fusion results or keep yes/no last message

Two scenarios are presented to compare the method keeping all the messages (first method) and the second method described in Section 3.6 which conserves and exchanges only the fused messages. Each method are tested using the discounting or the reinforcement process for messages agings.

Scenario n^{o}3 Three vehicles $v1$, $v2$ and $v3$ are considered in addition to a vehicle v.

This scenario proceeds as follows:

- a vehicle $v1$ creates a message $M1$ confirming the accident;
- vehicle $v1$ shares its message with vehicles $v2$ and $v3$;
- vehicle $v2$ creates a message $M2$ confirming the accident;
- vehicle $v2$ shares its database with v;
- vehicle $v3$ creates a message $M3$ confirming the accident;
- vehicle $v3$ shares its database with v.

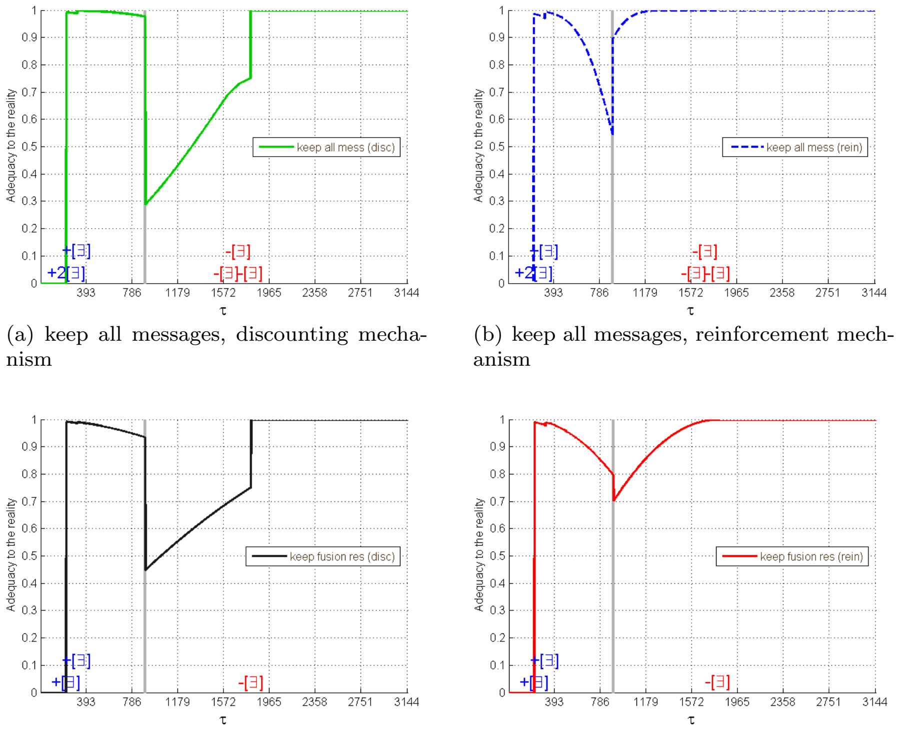
The database of the vehicle v depends on the method used:

- with the method where all messages are kept in vehicle database: vehicle v database contains $M1$ and $M2$, then $M1$, $M2$ and $M3$;
- with the method where only a synthesis message is kept in vehicle database: vehicle v database contains the fusion result message received from $v2$ (fusion of $M1$ and $M2$), then at the reception of the fusion result contained in $v3$ (fusion of $M1$ and $M3$), v applies an aging mechanism to both fusion results and combine them using the cautious rule of combination and keep in its database only the final fusion result.

The adequacy to the reality of the vehicle v is illustrated in Figure 6.

The method where vehicle keeps all messages in its database gives better results: before accident disappearance, the discounting mechanism gives the best result; then, the reinforcement mechanism gives the best result. This is due to the following reasons:

- the result of the conjunctive combination is higher than (or equal to) the result of the cautious conjunctive combination;
- discounting or reinforcing messages before combining them gives a higher result than combining messages before applying an aging mechanism.



(a) keep all messages, discounting mechanism

(b) keep all messages, reinforcement mechanism

(c) keep fusion result, discounting mechanism

(d) keep fusion result, reinforcement mechanism

Fig. 6 Adequacy to the reality of the information contained in vehicle v for each method during scenario n^{o}3

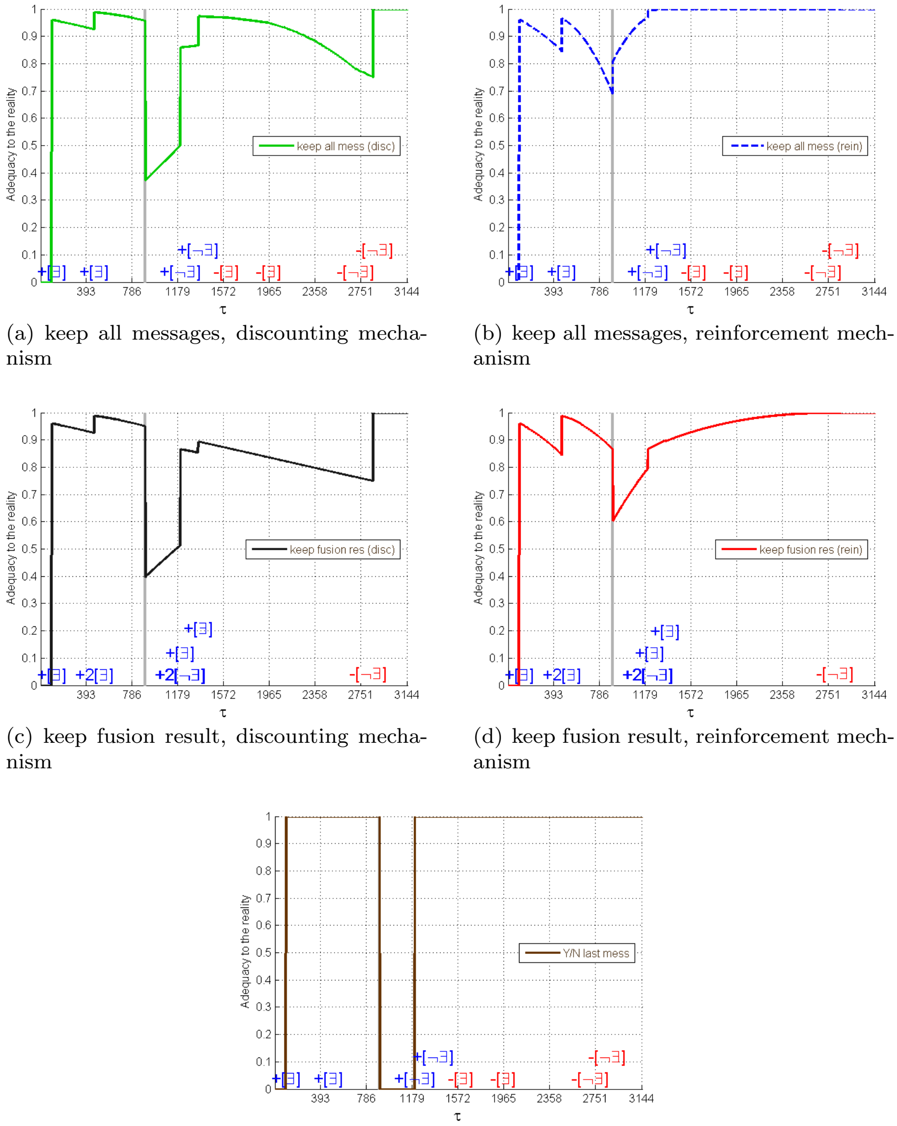
Scenario n^{o}4 In this scenario, previous methods are also compared with a very simple method called *y/n last mess* for short, where vehicles communicate simple messages telling *yes* or *no* an event exists; for each event, they keep in their databases only the last message. As all the drivers tell the truth (all the messages carry a true information), the last message is always the most recent true information.

In this scenario, a vehicle receives from distinct sources two messages confirming the accident, and two messages denying it after its disappearance.

Adequacy to the reality of the vehicle for each method is illustrated in Figure 7.

This scenario shows that after receiving messages denying the accident, the method where all messages are kept in vehicle database is still the best method when using reinforcement mechanism.

The result of the trivial method *y/n last mess* shows that this method does not allow predicting the accident disappearance, and gives a bad result after accident disappearance until the reception of a first message denying the



(a) keep all messages, discounting mechanism

(b) keep all messages, reinforcement mechanism

(c) keep fusion result, discounting mechanism

(d) keep fusion result, reinforcement mechanism

(e) yes/no last message

Fig. 7 Adequacy to the reality of each method: Scenario n°4

event. It means that even in an ideal not-very-real situation where drivers tell always the truth and messages are always received, a simple solution is not so evident.

5 An example of implementation with Android smartphones using Bluetooth transmissions

The purpose of this section is to illustrate how the first model presented in Section 3 can be implemented in Android smartphones with Bluetooth connections to illustrate a simple concrete case of use and to demonstrate that smartphones can manage belief functions calculations.

The choice of the Bluetooth technology is purely a matter of programming convenience. Data appearing in the following figures have been obtained on foot. Indeed no real tests were possible by car (or only at very slow speed) due to the fact that, with the Bluetooth version used by the authors, connection times were too long to allow smartphones to exchange data with cars passing on opposite sides on the road. As pointed out in the introduction of the article, the choice of the transmission technology is out of the scope of this article, the work presented here only concerns the application layer.

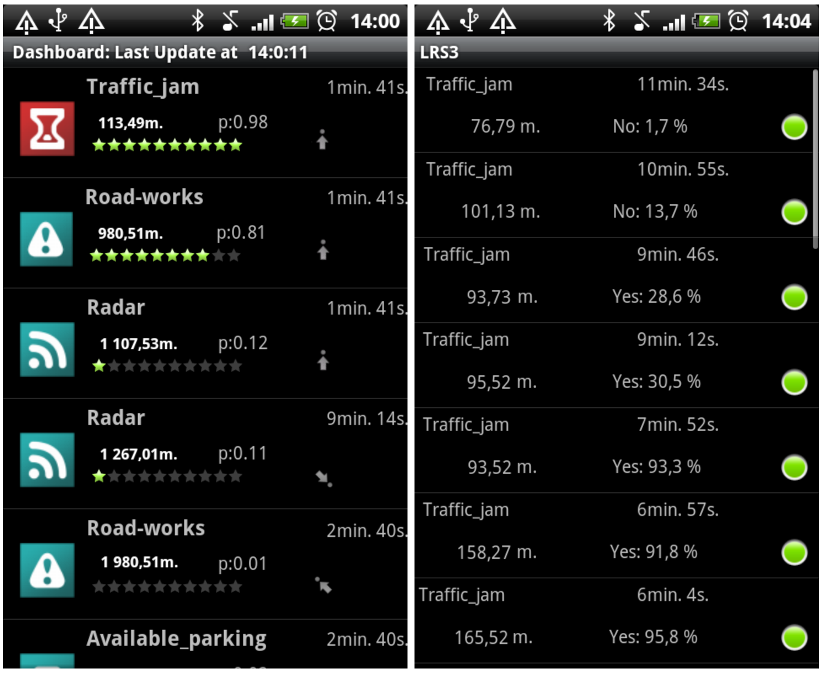
The following paragraphs detail both functional and technical aspects in relation with the implementation of the application and its possible services.

5.1 Application User Interfaces

As illustrated in Figure 3, the driver represents the unique actor of the system. He/She might play both sender and receiver roles, messages being periodically propagated "*broadcasted*" to the drivers within neighborhood coverage.

Application User Interfaces are based on four main screens:

- *Dashboard* (Figure 8): summarizes the description of the current vehicle environment. It lists a set of events sorted by their distance to the vehicle (Figure 8(a)). In addition to the event type, each list element shows the distance which is the average of the distances of the messages composing this event, the orientation of the event which is also the average of messages orientations and its pignistic probability which is the result of the aggregation of all the messages describing the event (cf Section 3.5). The time displayed near an event is the age of the most recent message in the list of messages constituting this event. For example, from Figure 8(a), there is a traffic jam in front of the vehicle at about 113 meters with a probability .98, and the most recent message concerning this event was received 1mn and 41s ago.
  The dashboard event list is periodically updated, taking into account time, new location, and received messages. Besides, the user have the possibility to launch the update manually. A detailed event view is also available. For each event, the set of belonging messages is made accessible by a simple selection of an event from dashboard. For example, a pressure on the first event "Traffic_jam" on the dashboard illustrated in Figure 8(a), gives the result illustrated in Figure 8(b) where messages corresponding to this event can be observed. For each message, the distance between the vehicle and the event, the age of the message and the value of the mass function are



Dashboard: Last Update at 14:0:11

Traffic_jam 1min. 41s.
113,49m. p:0.98

Road-works 1min. 41s.
980,51m. p:0.81

Radar 1min. 41s.
1 107,53m. p:0.12

Radar 9min. 14s.
1 267,01m. p:0.11

Road-works 2min. 40s.
1 980,51m. p:0.01

Available_parking 2min. 40s.

LRS3

Traffic_jam 11min. 34s.
76,79 m. No: 1,7 %

Traffic_jam 10min. 55s.
101,13 m. No: 13,7 %

Traffic_jam 9min. 46s.
93,73 m. Yes: 28,6 %

Traffic_jam 9min. 12s.
95,52 m. Yes: 30,5 %

Traffic_jam 7min. 52s.
93,52 m. Yes: 93,3 %

Traffic_jam 6min. 57s.
158,27 m. Yes: 91,8 %

Traffic_jam 6min. 4s.
165,52 m. Yes: 95,8 %

(a) Main dashboard giving information to the driver

(b) After having selected the first event "Traffic_jam" on the dashboard

Fig. 8 Application dashboard.

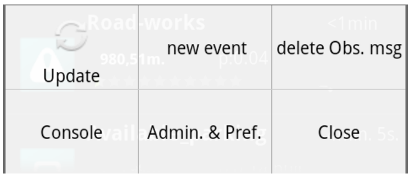
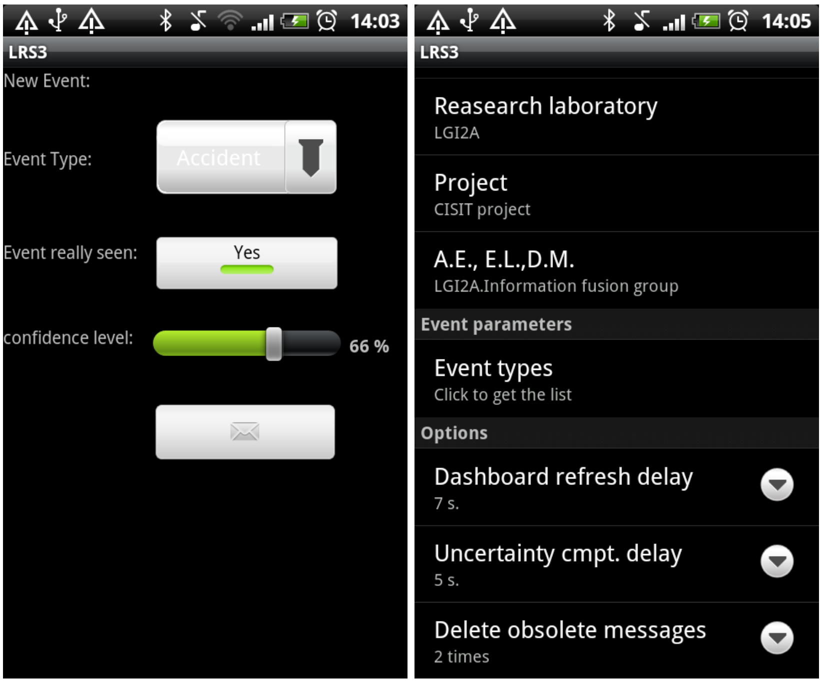


Fig. 9 Application menu

given. Each driver gives a simple mass function, then only masses on "yes, the event is present on the road at this time at my position" or "no it does not exist at this time at my position" are given (the complementary to one being given to Ω).

- *New event*: accessible via the menu interface (Figure 9), the new event interface (Figure 10(a)) affords to drivers the possibility to point out and report a new message describing an event. Drivers choose first the type of the event, then the purpose of the message: "yes, the event is present on the road at this time at my position" or "no it does not exist at this time at my position", and finally their degree of belief (in yes or no, the com-



14:03

LRS3

New Event:

Event Type: Accident

Event really seen: Yes

confidence level: 66 %

14:05

LRS3

Reasearch laboratory
LGI2A

Project
CISIT project

A.E., E.L.,D.M.
LGI2A.Information fusion group

Event parameters

Event types
Click to get the list

Options

Dashboard refresh delay
7 s.

Uncertainty cmpt. delay
5 s.

Delete obsolete messages
2 times

(a) Interface allowing drivers to send messages about an event

(b) Admin & Pref. interface

Fig. 10 Other user interfaces

plementary being given to Ω). Once the message created, it will be added to the database and periodically propagated to the neighboring vehicles.

- *Console*: is the user interface which traces application evolution and provides more detailed information about application components life cycle. Since there are several threads started by the application, the monitoring of the application is required. The console could be seen as an interactive log file. Such an interface is not of a great interest of final users but remains a considerable issue for debugging or/and experimentation.
- *Admin & Pref.* (Figure 10(b)): allows the user to manage event types as well as the configuration and the update of application parameters.

5.1.1 Software architecture: main application entities

Figure 11 illustrates the information system entities managed by the application. Message is the main entity. Each message belongs to a unique event. Messages classification is based on the event type and proximity indicators like location and time. The accuracy of information stream exchanged throughout messages are assessed, aggregated and merged according to their respective mass. For example, if and event is described by two messages, the respective

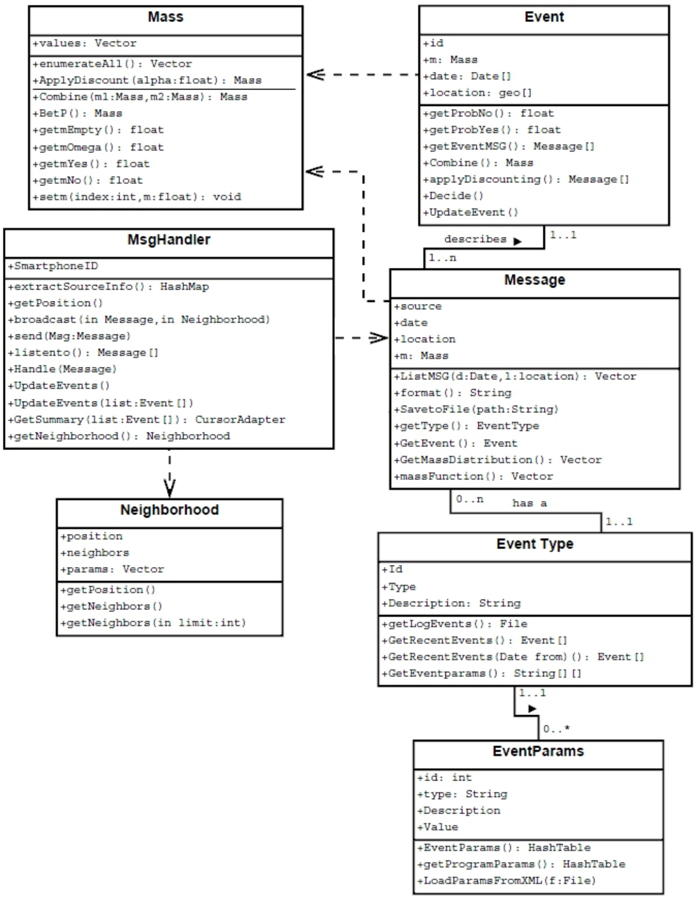


Fig. 11 Main application entities

masses are, first discounted, then combined and finally event consistency is assigned as a probability. Message handler class refers to the set of handler, periodic handler and procedures applied to both message and event databases. Finally, neighborhood entity denotes the set of devices discovers within neighborhood. We should note that the diagram illustrated by Figure 11, only covers representative functional entities. The class diagram takes into account more detailed and technical aspects of the application.

The application evolves through three states:

- Listening: the associated thread listen and waits for incoming connection. Once the connection is established, the current state changes to *receive* state.
- Receive: after state switching, a socket is created and the input stream is redirected. Then, binary data is read until the end of the stream. Binary data is converted to object messages. Next they are added to database.

- Send: Send state is used to propagate messages to neighboring smart phones. All non obsolete messages are sent to the connected devices.

6 Conclusion and future work

In this article, two models based on belief functions with two different message aging systems have been presented to manage uncertain messages informing drivers about events on the road in vehicular networks. A Matlab™ simulator has been developed to test them, and an implementation using Android smartphones have also been exposed to illustrate the feasibility of a concrete realization of these models.

One way to improve the method consists in trying to determine the age of an event from the series of messages related to it. It has indeed been observed that ideally messages created at the end of an event or after its disappearance should be reinforced towards the non existence of the event while messages confirming an event and created before its end should be kept. The knowledge of the beginning and the end of an event may significantly improve the results.

Another improvement concerns the spatial aspect of an event. In the models presented in this article an event takes place at a location ℓ which is sufficient for an accident but maybe not for a traffic jam which may spread to a large area. Likewise an accident can cause a traffic jam so the relations between types of events may also been considered.

Acknowledgements This work has been financed by the French region Nord-Pas de Calais under the project CISIT (Campus International pour la Sécurité et l'Intermodalité des Transports).

The authors are very grateful to the VESPA team, in particular Thierry Delot and Sylvain Lecomte from LAMIH Laboratory, Univ. of Valenciennes, for having helped them in their developments.

The authors would like to thank the anonymous reviewers for their valuable comments which have helped them to improve the clarity and the quality of this article.

References

1. A. Boukerche, Performance comparison and analysis of ad hoc routing algorithms, IEEE International Conference on Performance, Computing and Communications, pp. 171–178, (2001)
2. N. Cenerario, T. Delot, S. Ilarri, Dissemination of information in inter-vehicle ad hoc networks, IEEE Intelligent Vehicles Symposium (IV'2008), IEEE Computer Society, Eindhoven (The Netherlands), pp. 763-768, June (2008)
3. M. Bou Farah, D. Mercier, É. Lefèvre, F. Delmotte, Towards a robust exchange of imperfect information in inter-vehicle ad-hoc networks using belief functions, IEEE Intelligent Vehicles Symposium, IV'2011, pp. 436-441, Baden-Baden, Germany, June 5-9 (2011)
4. V. Cherfaoui, T. Denoeux, Z.L. Cherfi, Distributed data fusion: application to confidence management in vehicular networks, International Conference on Information Fusion (FUSION'2008), pp. 846–853, Germany (2008)
5. S. Corson, J. Macker, Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations, RFC 2501 (1999)

6. B. Defude, T. Delot, S. Ilarri, J-L Zechinelli-Martini, N. Cenerario, Data aggregation in VANETs: the VESPA approach, Mobiquitous'08 - First International Workshop on Computational Transportation Science (IWCTS), ACM digital library, Dublin, Ireland (2008)
7. L. Delgrossi, T. Zhang, Vehicle Safety Communications: Protocols, Security, and Privacy, John Wiley & Sons, Inc., Hoboken, NJ, USA (2012).
8. T. Delot, N. Cenerario, S. Ilarri, Estimating the relevance of information in inter-vehicle ad hoc networks, International Workshop on Sensor Network Technologies for Information Explosion Era (SeNTIE'08), IEEE Computer Society, pp. 151–158, Beijing, China (2008)
9. F. Delmotte, P. Smets, Target identification based on the transferable belief model interpretation of Dempster-Shafer model, IEEE Transactions on Systems, Man and Cybernetics A, vol. 34, pp. 457–471 (2004)
10. T. Denœux, Conjunctive and disjunctive combination of belief functions induced by non distinct bodies of evidence, Artificial Intelligence, vol. 172, issue 2-3, pp. 234-264, February (2008).
11. European CAR 2 CAR Communication Consortium, `http://www.car-to-car.org`
12. ITS in Japan, `http://www.mlit.go.jp/road/ITS/`
13. X. Hong, K. Xu, M. Gerla, Scalable routing protocols for mobile ad hoc networks xiaoyan, IEEE Network Magazine, vol. 16, pp. 11-21 (2002)
14. D. Mercier, T. Denœux, M.-H. Masson, Belief function correction mechanisms, Studies in Fuzziness and Soft Computing, B. Bouchon-Meunier et al. (Eds.), vol. 249, pp. 203–222, January (2010).
15. D. Mercier, É. Lefèvre, F. Delmotte, Belief functions contextual discounting and canonical decompositions, International Journal of Approximate Reasoning, vol.53, issue 2, pp. 146-158 (2012)
16. P. Morsink, R. Hallouzi, I. Dagli, C. Cseh, L. Schafers, M. Nelisse, D.D. Bruin, CarTalk 2000: Development of a cooperative ADAS based on vehicle-to-vehicle communication, World Congress and Exhibition on Intelligent Transport Systems and Services, Madrid, Spain, November (2003)
17. T. Nadeem, S. Dashtinezhad, C. Liao, L. Iftode, TrafficView: Traffic data dissemination using car-to-car communication, ACM Mobile Computing and Communications Review (MC2R), Special Issue on Mobile Data Management, vol. 8, issue 3, pp. 6–19, July (2004)
18. D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, W. Schulz, CarTALK 2000: safe and comfortable driving based upon inter-vehicle-communication, IEEE Intelligent Vehicle Symposium, vol. 2, pp. 545–550 (2002)
19. B. Ristic, P. Smets, Belief function theory on the continuous space with an application to model based classification, International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'04, pp. 4–9 (2004)
20. E.M. Royer, C.-K. Toh, A review of current routing protocols for ad hoc mobile wireless networks, IEEE Personal Communications, vol 6, issue 2, pp. 46-55 (1999)
21. G. Shafer, A Mathematical Theory of Evidence, Princeton University Press (1976)
22. P. Smets and R. Kennes, The Transferable Belief Model, Artificial Intelligence, vol. 66, pp. 191–243 (1994)
23. P. Smets, The Canonical Decomposition of a Weighted Belief, International Joint Conferences on Artificial Intelligence, pp. 1896–1901 (1995).
24. P. Smets, The Transferable Belief Model for quantified belief representation, handbook of defeasible reasoning and uncertainty management systems, D. M. Gabbay and Ph. Smets (Ed.), Kluwer Academic Publishers, Dordrecht, The Netherlands, vol. 1, pp. 267–301 (1998)
25. P. Smets, Decision making in the TBM: the necessity of the pignistic transformation, International Journal of Approximate Reasoning, vol. 38, issue 2, pp. 133-147 (2005)
26. US Department of Transportation RITA ITS program, `http://www.rita.dot.gov/`