RC23301 (W0408-092) August 19, 2004
Computer Science

IBM Research Report

Safety Driver Manager

Dimitri Kanevsky, Barbara Churchill, Alex Faisman,
David Nahamoo, Roberto Sicconi
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Safety Driver Manager

Dimitri Kanevsky , Barbara Churchill, Alex Faisman, David Nahamoo, Roberto Sicconi

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598
kanevsky@us.ibm.com

Abstract

Telematics services in cars (like navigation, cellular telephone, internet access) are becoming increasingly popular, but they may distract drivers from their main driving tasks and negatively affect driving safety.
This paper addresses some aspects of voice user interface in cars, as a mechanism to increase driver safety. Voice control becomes more efficient in reducing driver distraction if drivers can speak commands in a natural manner rather than having to remember one or two variants supported by the system. In this paper we discuss some ways to increase naturalness. Computers in cars are usually not very powerful due to cost considerations. Low CPU resources are a limiting factor for embedded speech solutions. Another aspect of this paper is a recently introduced novel solution for using a speech interface to reduce driver drowsiness and prevent a driver from falling asleep. All driver activities in cars (driving, talking over telephone, controlling Telematics devices, etc.) contribute to driver workload. Designing workload management in a user interface is a difficult task. In our paper we analyze some aspects of this problem. Finally, we introduce the idea of a distributed user interface between cars. It is well known that the safety of a driver depends not only on the driver himself but on the behavior of other drivers nearby. Therefore sharing some information about other cars and driver conditions could lead to increased driving safety. For example, if a driver in a nearby car is listening to an e-mail message or has had a high number of traffic accidents in the past, this “heightened risk” information could be sent anonymously to the workload manager in another car. The workload manager would then adjust risk factors in its safety assessment of the environment . In response to the heightened risk caused by the offending car, this workload manager may prevent the telephone from ringing or interrupt a dialog between the driver and a car system in other, nearby cars who are at higher risk because of the nearby “offending” car.

1. Introduction

Studies of road safety found that human error was the sole cause in more than half of all accidents (see for example, [16]. One of the reasons why humans commit so many errors lies in the inherent limitation of human information processing ([7]). With the increase in popularity of Telematics services in cars (like navigation, cellular telephone, internet access) there is more information that drivers need to process and more devices that drivers need to control that might contribute to additional driving errors. This paper is devoted to a discussion of these and other aspects of driver safety.

2. Voice control interface

One of the ways to address driver safety concerns is to develop an efficient system that relies on voice instead of hands to control Telematics devices. It has been shown in various experiments that well designed voice control interfaces can reduce a driver’s distraction compared with manual control situations (see for example [14] or [5]). One of the ways to reduce a driver’s cognitive workload is to allow the driver to speak naturally when interacting with a car system (e.g. when playing voice games, issuing commands via voice). It is difficult for a driver to remember a complex speech command menu (e.g. recalling specific syntax, such as "What is the distance to JFK?" or "Or how far is JFK?" or "How long to drive to JFK?" etc.). This fact led to the development of Conversational Interactivity for Telematics (CIT) speech systems at IBM Research.. CIT speech systems can significantly improve a driver-vehicle relationship and contribute to driving safety. But the development of full fledged Natural Language Understanding (NLU) for CIT is a difficult problem that typically requires significant computer resources that are usually not available in local computer processors that car manufacturers provide for their cars. To address this, NLU components should be located on a server that is accessed by cars remotely or NLU should be downsized to run on local computer devices (that are typically based on embedded chips). Some car manufacturers see advantages in using upgraded NLU and speech processing on the client in the car, since remote connections to servers are not available everywhere, can have delays, and are not robust. Our department is developing a “quasi-NLU” component - a “reduced” variant of NLU that can be run in CPU systems with relatively limited resources. It extends concepts described in the paper [3]. In our approach, possible variants for speaking commands are kept in special grammar files (one file for each topic or application). When the system gets a voice response, it searches through files (starting with the most relevant topic). If it finds an appropriate command in some file, it executes the command. Otherwise the system executes other options that are defined by a Dialog Manager (DM) . The DM component is a rule based sub-system that can interact with the car and external systems (such as weather forecast services, e-mail systems, telephone directories, etc.) and a driver to reduce task complexity for the NLU system. The following are examples of conversations between a driver and DM that illustrate some of tasks that an advanced DM should be able to perform:

1. *Ask questions (via a text to speech module) to resolve ambiguities:*
- (Driver) Please, plot a course to Yorktown
- (DM) Within Massachusetts?
- (Driver) No, in New York

2. *Fill in missing information and remove ambiguous references from context:*
- (Driver) What is the weather forecast for today?
- (DM) Partly cloudy, 50% chance of rain
- (Driver) What about Ossining?
- (DM) Partly sunny, 10% chance of rain

(The DM assumes that the driver means Yorktown, NY, from the earlier conversational context. Also, when the driver asks the inexplicit question "What about Ossining?" it assumes that the driver is still asking about weather.)

3. *Manage failure and provide contextual, failure- dependent help and actions*
- (Driver) When will we get there?
- (DM) Sorry, what did you say?
- (Driver) I asked when will we get there.

The quality of this "quasi-NLU" module depends on how well phrases in files cover multiple ways to speak commands. It is difficult to collect sufficient data that fully represents all possible ways that users might interact with a CIT system. No matter how large the data collection is, some users will produce some phrases that are not represented in the collected data nor in grammars that are developed from this data. Currently this issue is addressed by a convenient user interface that is provided by a research prototype of a CIT framework, that allows users to easily add new phrases and commands if they find that the system
does not understand the phrases they are using. Fig. 1 shows a snapshot of an interface that allows users to write new phrases and commands. The CIT interface is a special case of a general interactivity system that is developed in the ViaScribe speech interactive framework ([2]). A high view of the architecture of the CIT framework is shown in Fig. 2, in which Speech Engines (SE), Text To Speech (TTS), audio capturing/recording engines and some of the other components have been developed using C/C++. Various Dialog Managers (DM) have been developed using Java. The communication is provided by an Engine Manager component that is designed as a message bus, to which other parts of the framework can be connected. A workload manager (WM) receives sensor data from CIT Dashboard and other data and provides means to calculate cognitive weights. In the future, the problem of instantaneous data collection could be dealt systematically by creating a learning transformation system (LT). Examples of LT tasks are as follows:

- Monitor driver and passenger actions in the car's internal and external environments across a network;
- Extract and record the Driver Safety Manager relevant data in databases;
- Generate and learn patterns from stored data;
- Learn from this data how Safety Driver Manager components and driver behavior could be improved and adjusted to improve Driver Safety Manager performance and improve driving safety.

In particular, LT should be able to modify NLU components, such as files which include typical phrases that are linked with commands. For example, LT could add new phrases to NLU files that it finds from some drivers' dialogs or from more sophisticated automatic language models and NLU processors. If the number of phrases in some file become very large (which might lead to increased speech recognition error rates), then LT could split files by topics and adapt or create new grammars for domains related to such created topics. Examples of some technology that can be used for such topic identification are provided in [9]. When a sufficient number of phrases has accumulated, then statistical language models can be created from these corpora and augmented with grammar-based processors. A wide range of known mechanisms may be employed to promote interactions of LT with drivers, such as disclosed in [10] . On the other hand, the adaptation of Safety Driver Manager components (e.g. NLU, speech recognition, language models) related to similarities between users may also be carried out in any of a wide range of suitable methods, including those described in [11] and [12].

3. Embedded speech recognition

Car computers are usually not very powerful due to cost considerations. The growing necessity of the conversational interface demands significant advances in processing power on the one hand, and speech and natural language technologies on the other. In particular, there is significant need for a low-resource speech recognition system that is robust, accurate, and efficient. An example of a low-resource system that is executed by a 50 DMIPS processor, augmented by 1 MB or less of DRAM can be found in [2]. In what follows we give a brief description of the IBM embedded speech recognition system that is based on the paper [4]. Logically a speech system is divided into three primary modules: the front-end, the labeler and the decoder. When processing speech, the computational workload is divided approximately equally among these modules. In this system the front-end computes standard 13-dimensional mel-frequency cepstral coefficients

(MFCC) from 16-bit PCM sampled at 11.025 KHz. Front-End Processing Speech samples are partitioned into overlapping frames of 25 ms duration with a frame-shift of 15 ms. A 15 ms frame-shift instead of the standard 10 ms frame-shift was chosen since it reduces the overall computational load significantly without affecting the recognition accuracy. Each frame of speech is windowed with a Hamming window and represented by a 13 dimensional MFCC vector. We empirically observed that noise sources, such as car noise, have significant energy in the low frequencies and speech energy is mainly concentrated in frequencies above 200 Hz. The 24 triangular mel-filters are therefore placed in the frequency range [200Hz - 5500 Hz], with center frequencies equally spaced in the corresponding mel-frequency scale. Discarding the low frequencies in this way improves the robustness of the system to noise. The front-end also performs adaptive mean removal and adaptive energy normalization to reduce the effects of channel and high variability in the signal levels respectively. The labeler computes first and second differences of the 13-dimensional cepstral vectors, and concatenates these with the original elements to yield a 39-dimensional feature vector. The labeler then computes the log likelihood of each feature vector according to observation densities associated with the states of the system's HMMs. This computation yields a ranked list of the top 100 HMM states. Likelihoods are inferred based upon the rank of each HMM state by a table lookup ([1]). The sequence of rank likelihoods is then forwarded to the decoder. The system uses the familiar phonetically-based, hidden Markov model (HMM) approach. The acoustic model comprises context-dependent sub-phone classes (allophones). The context for a given phone is composed of only one phone to its left and one phone to its right. The allophones are identified by growing a decision tree using the context-tagged training feature vectors and specifying the terminal nodes of the tree as the relevant instances of these classes. Each allophone is modeled by a single-state Hidden Markov Model with a self loop and a forward transition. The training feature vectors are poured down the decision tree and the vectors that collect at each leaf are modeled by a Gaussian Mixture Model (GMM), with diagonal covariance matrices to give an initial acoustic model. Starting with these initial sets of GMMs several iterations of the standard Baum-Welch EM training procedure are run to obtain the final baseline model. In our system, the output distributions on the state transitions are expressed in terms of the rank of the HMM state instead of in terms of the feature vector and the GMM modeling the leaf. The rank of an HMM state is obtained by computing the likelihood of the acoustic vector using the GMM at each state, and then ranking the states on the basis of their likelihoods. The decoder implements a synchronous Viterbi search over its active vocabulary, which may be changed dynamically. Words are represented as sequences of context-dependent phonemes, with each phoneme modeled as a three-state HMM. The observation densities associated with each HMM state are conditioned upon one phone of left context and one phone of right context only. A discriminative training procedure was applied to estimate the parameters of these phones. MMI training attempts to simultaneously (i) maximize the likelihood of the training data given the sequence of models corresponding to the correct transcription, and (ii) minimize the likelihood of the training data given all possible sequences of models allowed by the grammar describing the task . The MMI estimation process that was used in this work is described in [6] and [15]. In 2001 , speech evaluation experiments yields improvement from 20% to 40% relatively depending on testing conditions (e.g. 7.6% error rate for 0 speed and 10.1% for 60 mph).

4. Driver drowsiness prevention – Artificial Passenger

Fatigue causes more than 240,000 vehicular accidents every year. Currently, drivers who are alone in a vehicle have access only to media such as music and radio news which they listen to passively. Often these do not provide sufficient stimulation to assure wakefulness. Ideally, drivers should be presented with external stimuli that are interactive to improve their alertness. Driving, however, occupies the driver's eyes and hands, thereby limiting most current interactive options. Among the efforts presented in this general direction, the invention [8] suggests fighting drowsiness by detecting drowsiness via speech biometrics and, if needed, by increasing arousal via speech interactivity. When the patent was granted in May 22, 2001, it received favorable worldwide media attention. It became clear from the numerous press articles and interviews on TV, newspaper and radio that Artificial Passenger was perceived as having the potential to dramatically increase the safety of drivers who are highly fatigued. It is a common experience for drivers to talk to other people while they are driving to keep themselves awake. The purpose of Artificial Passenger part of the CIT project at IBM is to provide a higher level of interaction with a driver than current media, such as CD players or radio stations, can offer. This is envisioned as a series of interactive modules within Artificial Passenger, that increase driver awareness and help to determine if the driver is losing focus. This can include both conversational dialog and interactive games, using voice only. The scenarios for Artificial Passenger currently include: quiz games, reading jokes, asking questions, and interactive books. In the Artificial Passenger (ArtPas) paradigm, the awareness-state of the driver will be monitored, and the content will be modified accordingly. Drivers evidencing fatigue, for example, will be presented with more stimulating content than drivers who appear to be alert. This could enhance the driver experience, and may contribute to safety.

The Artificial Passenger interaction is founded on the concept of psychological arousal. Most well known emotion researchers agree that arousal (high, low) and valence (positive, negative) are the two fundamental dimensions of emotion. Arousal reflects the level of stimulation of the person as measured by physiological aspects such as heart rate, cortical activation, and respiration. For someone to be sleepy or fall asleep, they have to have a very low level of arousal. There is a lot of research into what factors increase psychological arousal since this can result in higher levels of attention, information retention and memory. We know that movement, human voices and faces (especially if larger than life), and scary images (fires, snakes) increase arousal levels. We also know that speaking and laughing create higher arousal levels than sitting quietly. Arousal levels can be measured fairly easily with a biometric glove (from MIT), which glows when arousal levels are higher (reacts to galvanic skin responses such as temperature and humidity). The following is a typical scenario involving Artificial Passenger:

Imagine, driver "Joe" returning home after an extended business trip during which he had spent many late nights. His head starts to nod …

ArtPas: Hey Joe, what did you get your daughter for her birthday?
Joe (startled): It's not her birthday!
ArtPas: You seem a little tired. Want to play a game?
Joe: Yes.
ArtPas: You were a wiz at "Name that Tune" last time. I was impressed. Want to
try your hand at trivia?
Joe: OK.
ArtPas: Pick a category: Hollywood Stars, Magic Moments or Hall of Fame?
Joe: Hall of Fame.
ArtPas: I bet you are really good at this. Do you want the 100, 500 or 1000 dollar
level?
Joe: 500
ArtPas: I see. Hedging your bets are you?

By the time Joe has answered a few questions and has been engaged with the dynamics of the game, his activation level has gone way up. Sleep is receding to the edges of his mind. If Joe loses his concentration on the game (e.g. does not respond to the questions which Artificial Passenger asks) the system will activate a physical stimulus (e.g. verbal alarm). The Artificial Passenger can detect that a driver does not respond because his concentration is on the road and will not distract the driver with questions. On longer trips the Artificial Passenger can also tie into a car navigation system and direct the driver to a local motel or hotel.

5. Workload Manager

In this section we provide a brief analysis of the design of the workload management that is a key component of driver Safety Manager (see Fig. 3). An object of the workload manager is to determine a moment-to-moment analysis of the user's cognitive workload. It accomplishes this by collecting data about user conditions, monitoring local and remote events, and prioritizing message delivery. There is rapid growth in the use of sensory technology in cars. These sensors allow for the monitoring of driver actions (e.g. application of brakes, changing lanes), provide information about local events (e.g. heavy rain), and provide information about driver characteristics (e.g. speaking speed, eyelid status). There is also growing amount of distracting information that may be presented to the driver (e.g. phone rings, radio, music, e-mail etc.) and actions that a driver can perform in cars via voice control. The relationship between a driver and a car should be consistent with the information from sensors. The workload manager should be designed in such a way that it can integrate sensor information and rules on when and if distracting information is delivered. This can be designed as a "workload representational surface". One axis of the surface would represent stress on the vehicle and another, orthogonally distinct axis, would represent stress on the driver. Values on each axis could conceivably run from zero to one. Maximum load would be represented by the position where there is both maximum vehicle stress and maximum driver stress, beyond which there would be "overload". The workload manager is closely related to the event manager that detects when to trigger actions and/or make decisions about potential actions. The system uses a set of rules for starting and stopping the interactions (or interventions). It controls interruption of a dialog between the driver and the car dashboard (for example, interrupting a conversation to deliver an urgent message about traffic conditions on an expected driver route). It can use answers from the driver and/or data from the workload manager relating to driver conditions, like computing how often the driver answered correctly and the length of delays in answers, etc. It interprets the status of a driver's alertness, based on his/her answers as well as on information from the workload manager. It will make decisions on whether the driver needs additional stimuli and on what types of stimuli should be provided (e.g. verbal stimuli via speech applications or physical stimuli such as a bright light, loud noise, etc.) and whether to suggest to a driver to stop for rest. The system permits the use and testing of different statistical models for interpreting driver answers and information about driver conditions. The driver workload manager is connected to a driving risk evaluator that is an important component of the Safety Driver Manager. The goal of the Safety Driver Manager is to evaluate the potential risk of a traffic accident by producing measurements related to stresses on the driver and/or vehicle, the driver's cognitive workload, environmental factors, etc. The important input to the workload manager is provided by the situation manager whose task is to recognize critical situations. It receives as input various media (audio, video, car sensor data, network data, GPS, biometrics) and as output it produces a list of situations. Situations could be simple, complex or abstract. Simple situations could include, for instance: a dog locked in a car; a baby in a car; another car approaching; the driver's eyes are closed; car windows are closed; a key is left on a car seat; it is hot in a car; there are no people in a car; a car is located in New York City; a driver has diabetes; a driver is on the way home. Complex situations could include, for example: a dog locked is locked in a car AND it is hot in a car AND car windows are closed; a baby is in a car AND there are no people in a car; another car is approaching AND the driver is looking in the opposite direction; a key is left on a car seat AND a driver is in the midst of locking a car; the driver is diabetic AND has not taken a medicine for 4 hours. Abstract situations could include, for example: Goals: get to work, to cleaners, to a movie. Driver preferences: typical routes, music to play, restaurants, shops. Driver history: accidents, illness, visits. Situation information can be used by different modules such as workload, dialog and event managers; by systems that learns driver behavioral patterns, provide driver distraction detection, and prioritize message delivery. For example, when the workload manager performs a moment-to-moment analysis of the driver's cognitive workload, it may well deal with such complex situations as the following: Driver speaks over the phone AND the car moves with high speed AND the car changes lanes; driver asks for a stock quotation AND presses brakes AND it is raining outside; another car approaches on the left AND the driver is playing a voice interactive game.

The dialog manager may well at times require uncertainty resolution involving complex situations, as exemplified in the following verbal query by a driver: "How do I get to Spring Valley Rd?"

Here, the uncertainty resides in the lack of an expressed (geographical) state or municipality. The uncertainty can be resolved through situation recognition; for example, the car may be in New York State already (that is defined via GPS) and it may be known that the driver rarely visits other states. The concept associated with learning driver behavioral patterns can be facilitated by a particular driver's repeated routines, which provides a good opportunity for the system's "learning" habitual patterns and goals. So, for instance, the system could assist in determining whether drivers are going to pick up their kids in time by, perhaps, reordering a path from the cleaners, the mall, the grocery store, etc.

6. Privacy and social aspects

Addressing privacy concerns: The safety driver manager framework should be designed such that it will be straightforward for the application designers to protect the end user's privacy. This should include encryption of the message traffic from the vehicle, through a carrier's network, and into the service provider's secure environment, such that the driver's responses cannot be intercepted. This can be achieved through the use of IBM WebSphere Personalization Server or Portal Server, allowing the end user an interface to select options and choices about the level of privacy and/or the types of content presented. An example of such an option is the opportunity for drivers to be informed about the existence of the Artificial Passenger capability, with clear instructions about how to turn it off if they opt not to use it.

Addressing social concerns: The safety driver manager is being developed to enable service providers to enhance the end user's driving experience, and the system will be designed to ensure that it has this desired effect. The social impact of the system will be managed by making sure that users clearly understand what the system is, what the system can and cannot do, and what they need to do to maximize its performance to suit their unique needs. For example, in the Artificial Passenger paradigm the interaction can be customized to suit the driver's conversational style, sense of humor, and the amount of "control" that he/she chooses to leave to the Artificial Passenger system (e.g., some drivers might find it disconcerting if the Artificial Passenger system opens the window for them automatically; others might find this a key feature.). The system will include a learning module that detects and records customer feedback, e.g. if a driver does not laugh at certain type of jokes, then that type will not be presented. Positive feedback in one area (football scores from a driver's home town) leads to additional related content (baseball scores from the same town, weather, etc.). The social concerns associated with Artificial Passenger can be addressed by allowing the users to specify their desires and requirements through the subscriber management tools.

A general approach to privacy, social and legal issues in Telematics can be found in [13]. Some elements of this approach (e.g. Privacy Manager, Insurance) are reflected in Fig. 3.

7. Distributive user interface between cars

The safety of a driver depends not only on the driver himself but on the behavior of other drivers near him. Existing technologies can attenuate the risks to a driver in managing his/her own vehicle, but they do not attenuate the risks presented to other drivers who may be in "high risk" situations, because they are near or passing a car where the driver is distracted by playing games, listening to books or engaging in a telephone conversation. It would thus appear helpful at times to inform a driver about such risks associated with drivers in other cars. In some countries, it is required that drivers younger than 17 have a mark provided on their cars to indicate this. In Russia (at least in Soviet times), it was required that deaf or hard of hearing drivers announce this fact on the back of the window of his or her car. There is, then, an acknowledged need to provide a more dynamic arrangement to highlight a variety of potentially dangerous situations to drivers of other cars and to ensure that drivers of other cars do not bear the added responsibility of discovering this themselves through observation, as this presents its own risks. Information about driver conditions can be provided from sensors that are located in that car. The following are examples of the information about drivers that can affect driver conditions:

- mood (angry, calm, laughing, upset)
- physical conditions (tired, drowsy, sick, has chronic illnesses that can affect driving – like diabetes)
- attention (looking on a road or navigation map in a car, talking to a baby in a back sit, talking over telephone, listening to e-mail)
- driver profile (number of traffic accidents, age).

There can be several ways to assess this information . Driver overall readiness for safe driving can be evaluated by a safety manager in his/her car. It can be ranked by some metrics (e.g. on a scale from 1 to 5) and this evaluation can then be sent to the driver safety managers in nearby cars. Another way is that a driver manager in one car has access to information to a driver profile and from sensors in other cars. This second method allows individual car drivers to customize their priorities and use personal estimators for driving risks factors. For example, some one who is more worried about young drivers may request that this information be provided to his/her driver safety manager rather than an overall estimation of risk expressed as a single number. If a driver safety manager finds that there is additional risk associated with driver behavior in a car located nearby, it may prevent a telephone ringing or interrupt a dialog between the driver and a car system if. It can also advise someone who is calling a driver that that driver is busy and should not be disturbed at this time. The information can be sent anonymously to the driver safety manager in another car and this manager would then adjust the risk factor in its estimation of the surrounding environment for this car. This allows the system to address privacy concerns that drivers may have. One can also offer reduced insurance payments to a driver if s/he agrees to disclose information to other cars. Employers of fleet tracks may be particularly interested in this approach since it allows reduction in traffic accidents.

8. Conclusion

In the paper we suggested that such important issues related to a driver safety, such as controlling Telematics devices and drowsiness can be addressed by a special speech interface. This interface requires interactions with workload, dialog, event, privacy, situation and other modules. We showed that basic speech interactions can be done in a low-resource embedded processor and this allows a development of a useful local component of Safety Driver Manager. The reduction of conventional speech processes to low – resources processing was done by reducing a signal processing and decoding load in such a way that it did not significantly affect decoding accuracy and by the development of quasi-NLU principles. We observed that an important application like Artificial Passenger can be sufficiently entertaining for a driver with relatively little dialog complexity requirements – playing simple voice games with a vocabulary containing a few words. Successful implementation of Safety Driver Manager would allow use of various services in cars (like reading e-mail, navigation, downloading music titles etc.) without compromising a driver safety. Providing new services in a car environment is important to make the driver comfortable and it can be a significant source of revenues for Telematics. We expect that novel ideas in this paper regarding the use of speech and distributive user interfaces in Telematics will have a significant impact on driver safety and they will be the subject of intensive research and development in forthcoming years at IBM and other laboratories.

9. Acknowledgements. We would like to thank Sara Basson, Liam Comerford, Ponani Gopalakrishan, Steve Ross, Sally Lake, Paul Moskowitz, and Mahesh Viswanatha for useful discussions, advice and help in preparation of the paper.

10. References

[1] L.R. Bahl et al., "Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task," ICASSP 1995, vol.1, pp 41-44.).

[2] K. Bain, S. Basson, A. Faisman, D. Kanevsky "LIBERATED LEARNING and IBM ViaScribe: Accessibility through speech recognition submitted to IBM Systems Journal, Special Issue on Accessibility"

[3] Comerford L., Frank D., Gopalakrsihnan P., Gopinath R., and Sedivy J., "The IBM personal Speech Assistant", Proc. of the ICASSP 2001

[4] S. Deligne, E. Eide, R. Gopinath, D. Kanevsky, B. Maison, P. Olsen, H. Printz, J. Sedivy: "Low-resource speech recognition of 500-word vocabularies", Proceedings of EUROSPEECH 01, 2001.

[5] Gellatly, A.W., and Dingus, T.A. Speech recognition and automotive applications: using speech to perform in-vehicle tasks. Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting, (Santa Monica CA, 1998), Human Factors and Ergonomics Society, 1247-1251.

[6] P.S. Gopalakrishnan, D. Kanevsky, A. N'adas and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," IEEE Transactions on Information Theory, 37(1), January 1991

[7] Marc Green & John Senders, www.visualexpert.com/Resources/roadaccidents.html
[8] Kanevsky, D. and Zadrozny, W., U.S. Patent No. 6,236,968 , “Sleep prevention dialog based car system”, 2001.
[9] Kanevsky, D. and Yashchin, E., U.S. Patent No. 6,529,902, “Method and system for off-line detection of textual topical changes and topic identification via likelihood based methods for improved language modeling”, 2003.
[10] Kanevsky, D., Zlatsin, A. and Pickover, C., U.S. Patent No. 6,505,208, “Educational monitoring method and system for improving interactive skills based on participants on the network”, 2003.
[11] Kanevsky, D., Wolf, C. and Zadrozny, W., U.S. Patents: No. 6,484,136 , “Language model adaptation via network of similar users”, 2002.
[12] Kanevsky, D., Libal, V., Sedivy, J., and Zadrozny, W., U.S. Patents: No 6,442,519, “Speaker model adaptation via network of similar users”, 2002.
[13] Paul Moskowtiz, Ronald Perez, Moninder Singh, Jung-Mu Tang, “Mobile Networks and Applications”s Volume 9 issue 5, 2004. Some elements of this approach (e.g. Privacy Manager, Insurance)
[14] Marvin C. McCallum, John L. Campbell, Joel B. Richman and James L. Brown, "Speech Recognition and In-Vehicle Telematics Devices: Potential Reductions in Driver Distraction", published in International Journal of Speech Technology, v.7, N 1, 2004.
[15] Y. Normandin, "Hidden Markov Models, Maximum Mutual Information Estimation and the Speech Recognition Problem", PhD thesis, McGill University, Montreal, 1991.
[16] Treat, J. R., Tumbas, N. S., McDonald, S. T., Shinar, D., Hume, R. D., Mayer, R. E., Stanisfer, R. L. and Castellan, N. J. (1977) Tri-level study of the causes of traffic accidents. Report No. DOT-HS-034-3-535-77 (TAC).).

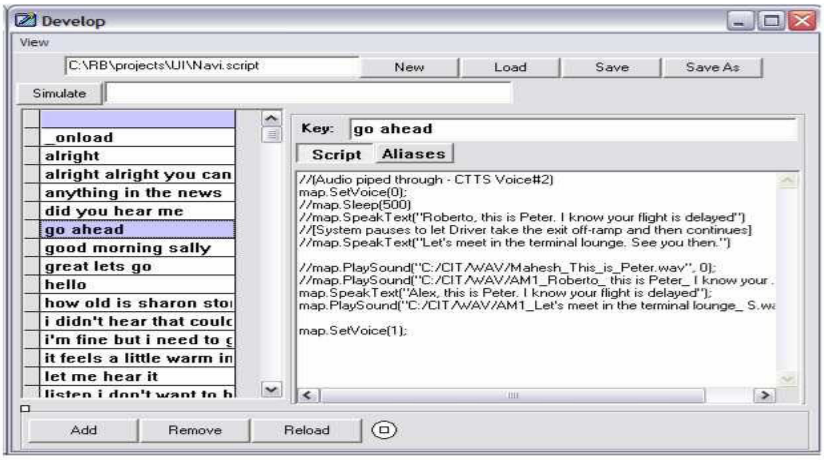

Fig. 1 User Interface to explore speech-based interactivity

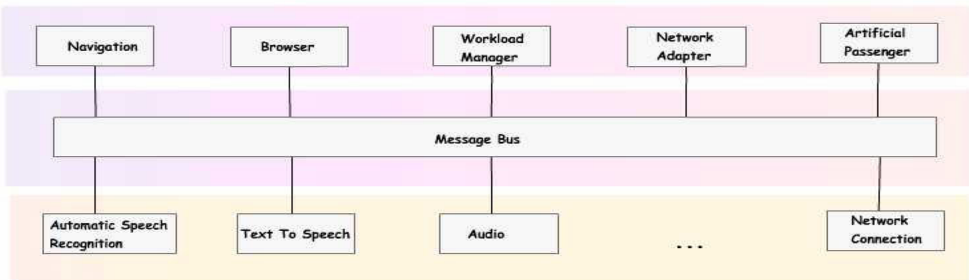

Fig. 2 Illustration of modular design of CIT Framework

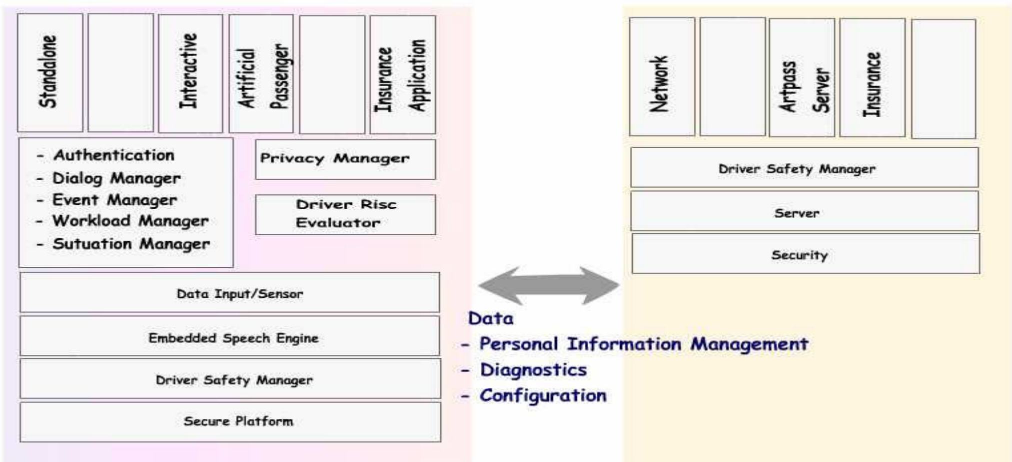

Fig. 3 Driver Safety Manager in a sample Telematics Platform