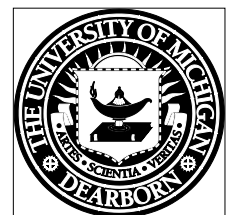

Developing In-Vehicle Database Management Techniques for Efficient Vehicular Applications



The University of Michigan-Dearborn
Henry W. Patton Center for Engineering
Education and Practice

Henry W. Patton Center for Engineering
Education and Practice
Annual Progress Report

Developing In-Vehicle Database Management Techniques for Efficient Vehicular Applications

(Project 2005/9)

By:

Qiang Zhu, Associate Professor
Brahim Medjahed, Assistant Professor

Department of Computer and Information Science

Table of Contents

Synopsis	iii
1. Background	1
2. Objectives.....	2
3. Approach.....	2
4. Results	4
5. Conclusions.....	5
6. Impact	
Educational.....	5
Industrial	5
7. Acknowledgement	6

Synopsis

Data management has become a crucial issue in achieving efficient vehicular functions/applications such as road navigation, music playing, and malfunctioning module diagnosis for a vehicle. At present, different applications adopt different proprietary ways to manage their data in the same vehicle, which suffers a number of drawbacks including lack of data sharing/integration, difficulty for information exchange, and inability for dynamic data updating. On the other hand, existing generic database management systems (DBMS) may not provide a good solution for in-vehicle data management since they are either heavy-weighted (e.g., the client-server DBMSs: Oracle, DB2 and SQL Server) or not designed for efficiently supporting a large number of applications in the same environment (e.g., the embedded DBMSs: ENCIRQ and PointBase Embedded). In this project, we develop appropriate techniques to address relevant challenges for in-vehicle data management. It is expected that an In-Vehicle Database Management System (IVDBMS) incorporating our techniques can not only efficiently support existing vehicular applications but also provide a platform for developing new vehicular applications in a fast and cost-effectiveway.

1. Background

As today's vehicles are equipped with more and more functions/applications (e.g., real time navigation, digital music playing, traveling log, and calendar), data in a vehicle has grown rapidly in terms of both the volume and types. How to efficiently manage such in-vehicle data is essential to warrant the success of vehicular applications.

Unfortunately, the solution adopted in current vehicles to manage their data is unsatisfactory. At present, each vehicular application possesses a private data management layer. The data management techniques employed in some vehicular applications are quite preliminary. It has a number of shortcomings. First of all, a lot of data owned by different vehicular applications cannot be shared. This is because an application does not provide a necessary application programming interface (API) and relevant database schema information for other vehicular applications. On the other hand, there are many useful vehicular applications that require information from multiple data sources. For example, the travel log feature for fleet vehicle needs location information from navigation and fuel level information from instrument cluster. An integrated in-vehicle database would make it possible to develop many such vehicular applications. Secondly, the current solution makes it very difficult to utilize different subsystems for an application from different suppliers since each application supplier provides a complete system as a black box. For example, Ford may prefer the navigation from supplier *X* with the map database from supplier *Y*, however, supplier *X* only uses a map database from supplier *Z* instead of *Y*. Although the systems from different suppliers may have different unique features, it is very difficult to integrate them into one system since the data management components that they are based on are different. Separating the data management layer from the rest of the application logic not only allows different suppliers to provide solutions to different aspects of an application, but also improves the overall performance of vehicular applications. The latter is because the duplication of some common data management programs (e.g., indexing) is eliminated by the adoption of a shared DBMS. Thirdly, the current solution does not efficiently support prompt dynamic information updates. Nowadays a vehicle can receive dynamic real-time (digital) information about relevant events/things (such as road constructions, weather conditions and new music releases) via a special radio channel or a wireless network connection. However, the current vehicular applications only update their databases periodically through updated CDs shipped by their suppliers rather than incorporating the dynamic information immediately into the databases. An in-vehicle DBMS can update the database in time based on available dynamic information. Fourthly, the current solution makes it difficult to exchange information with applications outside the vehicle. As part of a mobile computing environment, a vehicle is expected to exchange information with other vehicles and/or base stations. If different vehicular applications manage their own data, such interactions have to be handled module by module within the vehicle. An in-vehicle database would provide a single access point and a common access interface to make such interactions much easier. Finally, some useful information (such as personal profile, vehicle repairing history and calendar) is currently not fully available in a vehicle. An in-vehicle database would make more information available for new applications.

There are two types of DBMSs in the market: the enterprise ones (such as Oracle, DB2 and SQL Server) and the embedded ones (such as PointBase and ENCIRQ). However, they do not directly meet the needs of in-

vehicle database management. The enterprise DBMSs are very heavy-weighted and contain many components that are not needed for in-vehicle data management. Using such a DBMS would waste precious in-vehicle computing resources. The embedded DBMSs, on the other hand, were mainly designed for managing data for individual embedded devices. The components of such a system are selectively linked into an application, and it assumes no direct interaction with end users. Since a vehicle may have more data types and data volume, the embedded database system needs to be improved to handle more interactions and transactions.

2. Objectives

The main objective of this project is to develop appropriate in-vehicle data management techniques and conduct theoretical and empirical studies to evaluate their feasibility and effectiveness in vehicular applications. Several students will be trained with knowledge and experience in the field via the research. Researchers from both the university and Ford will have opportunities to exchange information and ideas on cutting-edge database management technologies and real-world issues.

3. Approach

There are different types of data, such as songs, maps, diagnostic codes, personal information and traveling logs, for vehicular applications. Such data may reside in multiple media such as CDs, iPod, hard-disks, USB drives, and palm devices. It may be represented in different formats (structured or unstructured). Since XML facilitates the integration of heterogeneous data, provides a consistent way to exchange information among applications and supports flexibility and extensibility, it is adopted to model in-vehicle data in our project. An XML schema capturing typical in-vehicle data has been designed. Since some in-vehicle data such as traveling logs and descriptive music information are well-structured and can be processed/managed efficiently using relational database techniques, we are also investigating relevant issues for a hybrid XML/relational database for vehicular applications.

The architecture of an in-vehicle database management system (IVDBMS) designed for our project is illustrated in Figure 1. We notice that, besides the typical techniques for query processing, concurrency control, metadata management, etc., an IVDBMS also requires special techniques to address unique issues for in-vehicle data management. Some of such techniques that we have studied are described as follows.

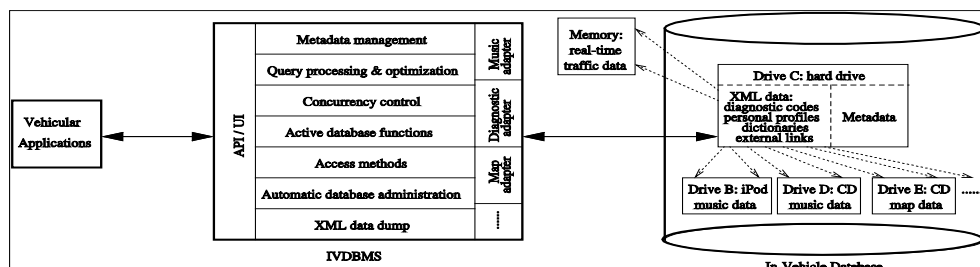


Figure 1.

Synchronization is one of the unique requirements for in-vehicle data management. Users/drivers of modern vehicles should be able to plug in their external data media such as CD/DVD's and MP3 players for relevant vehicular applications. The IVDBMS needs an effective mechanism to identify whether a plugged-in medium is new, old, or updated (with some insertions/deletions/modifications). When a new medium is plugged in, the IVDBMS extracts relevant descriptive information from the medium and stores it in a directory maintained by the system so that vehicular applications (e.g., a music player) can search information in the database including connected external data media. When an old medium is plugged in, the relevant old entries in the directory need to be re-activated, but no changes are needed. When an updated medium is plugged in, synchronization is required so that the information in the directory will reflect the updated medium. We have studied four synchronization strategies, i.e., the eager, on-demand, lazy and offline approaches. The strengths and weaknesses of each approach are analyzed.

Progressive query processing is another special issue for in-vehicle data management. Unlike a traditional query, a query Q for a vehicular application may be formulated in several steps (progressively). Such Q is called a progressive query, and each of its steps is called a step query SQ of Q . For example, assume a user wanted to select a song to play in a vehicle. He first asked the system to list all songs released in 2005. He found that there were too many such songs in the database. He then requested to narrow down the list by imposing a further condition on the genre. For such a progressive query, a user always formulates the next step query based on the result of the previous step. Hence a progressive query cannot be known beforehand, which raises new challenges for optimizing such queries. Figure 2 summarizes the different types of progressive queries (solid arrows are for external input tables).

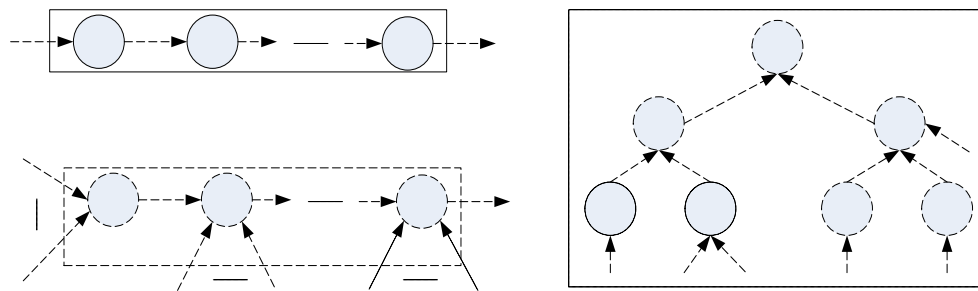


Figure 2.

To efficiently process a progressive query, we have developed a new index technique, called the collective index (see Figure 3). The key idea is to adopt an efficient scheme to transform indexes for the input table(s) of a step query into indexes for the result/output table of the step query, so that the next step query can be processed efficiently using the transformed indexes. Note that dynamically creating indexes from scratch for the input table of each step is infeasible due to large overhead. Our collective index consists of a collection of member indexes with some special structure to facilitate the ease of index maintenance/transformation. Each member index is a B+-tree except that its entries in the leaf nodes point to entries in a special router rather than point to tuples in the indexed table directly. Each tuple in the indexed table has one entry in the router,

and vice versa. Note that the leaf node entries from all member indexes that correspond to the same tuple will point to the same entry in the router. For a given step query, if a tuple in the input table satisfies the given query condition, it will be put in the result table. Without the collective index structure, to maintain a collection of traditional B+-trees, we would have to go through each individual index and make the relevant leaf node entry point to the new tuple. With the collective index, we only need to make the relevant entry in the router point to the tuple in the new result table, and then all member indexes have their relevant leaf node entries automatically point to the right tuple in the new table, which greatly reduces the index maintenance/transformation cost. To further improve performance of the collective index, we also employ a buffering strategy for reading router entries from a disk. Our experiments demonstrate that our collective index (CI) outperforms the consecutive linear scan method (CLS, i.e., applying the linear scan at each step) for processing progressive queries (see Figure 4).

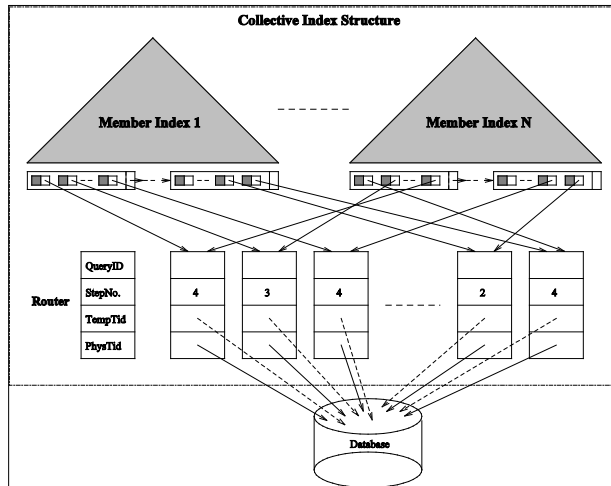


Figure 3.

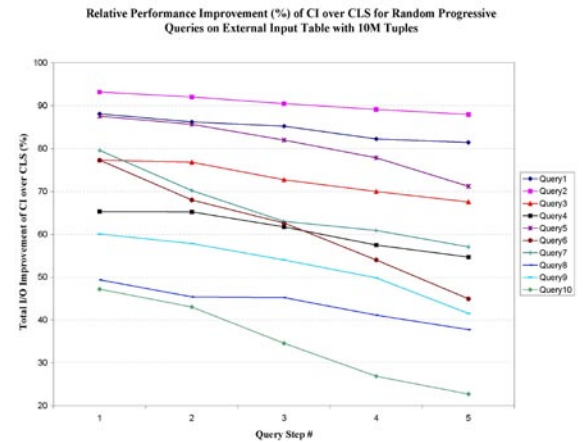


Figure 4.

The other relevant research issues that we are investigating include query processing over data streams and dynamic information superposition and update. Further issues will be studied in continuing projects.

4. Results

Research activities for the project have been conducted for one year. The research work has been well on its target. The following achievements have been made so far:

- A new collective indexing technique to efficiently process progressive queries required by vehicular applications has been developed.
- A framework to demonstrate essential components of an In-Vehicle Database Management System has been developed.
- An XML database schema to capture typical in-vehicle data has been designed.
- A set of synchronization techniques to handle various plug-in data devices have been proposed.

- A number of storage structures to store XML data in an IVDBMS have been studied.
- Experimental programs and data to evaluate relevant techniques have been obtained.
- Four relevant proposals to seek external funds from Ford Motor Company, National Science Foundation (NSF) and Microsoft were submitted. One award of \$242,582 from NSF was received.
- One poster entitled "Developing In-Vehicle Database Management Techniques for Efficient Vehicular Applications" (by Q. Zhu and B. Medjahed) was given at Technology Day 2006, sponsored by the Henry W. Patton Center for Engineering Education and Practice, at the University of Michigan-Dearborn.
- One technical paper entitled "The Collective Index: A Technique for Efficient Processing of Progressive Queries" (by Q. Zhu, B. Medjahed, A. Sharma and H. Huang) was written and submitted to a journal for publication.
- A number of graduate students obtained relevant research experience via their thesis work and course projects (see Impact section for details).

5. Conclusions

Data management has become a crucial issue in achieving efficient vehicular applications in modern vehicles. Although some existing techniques may be applied to solve some in-vehicle data management issues, new techniques are required to tackle special challenges raised by vehicular applications. An IVDBMS incorporating the new techniques developed in our project is expected to not only efficiently support existing vehicular applications but also provide a platform for developing new vehicular applications in a fast and cost-effective way.

6. Impact

The project has impact on both education and industries. Specifically,

- A. Educational Impact:** So far, (1) three graduate students in Advanced Database Systems (CIS 586) completed three relevant course projects; (2) twelve graduate students in Web Information Management (CIS 562) completed four relevant group course projects; (3) two graduate student research assistants worked on two Master theses on relevant research; (4) Advanced Database Systems (CIS 586) was revised to incorporate a relevant new chapter; (5) Web Information Management (CIS 562) was created to teach essential knowledge. We expect that more course projects will be completed for the research and several students will be trained in the area.
- B. Industrial Impact:** (1) Research of the project well fits the needs of upcoming modern vehicles, including utilizing large in-vehicle hard drive storage, supporting complex vehicular applications accessing data from multiple sources, and participating in mobile and Web-service environments; (2) The techniques to be developed in this project have a significant impact on achieving efficient and effective vehicular applications; (3) The project will provide a bridge between academic research and industrial applications. The university researchers will learn the practical requirements and knowledge, and the industrial

collaborators will understand the research state in the area. (4) The project will train qualified software engineers in databases and their applications for industry.

6. Acknowledgment

This research was sponsored by NSF (under grant #: CNS-0521142) and the Henry W. Patton Center for Engineering Education and Practice at the University of Michigan-Dearborn. We would like to thank Henry Huang, Vladimir Rasin, Edward Nelson, Venkatesh Prasad, Robert W. Hildebrand, John J. Cristiano, and Donna Goddard for their valuable suggestions, support and help for the project.