学位報告 4

報告番号	※甲　　第　　号

主　論　文　の　要　旨

論文題目　車載データストリーム管理システムのランタイムコード生成技術

氏　　名　勝沼　聡

論 文 内 容 の 要 旨

近年，車載システムでは，プリクラッシュセーフティ・システムや ACC（Adaptive Clause Control），車線維持支援システムなど，車両の状態や周辺状況を判断し，ドライバへの警告や自動制御により運転の支援を行うシステム（以下，運転支援システム）が登場している．運転支援システムでは，周辺の物体を検知するためにミリ波レーダやレーザレーダ，カメラを始め，車輪速センサ，加速度センサ，位置検出センサなど多様なセンサを複数搭載する．また車々間通信や路車間通信の利用も検討されており，車と車あるいは，車と道路のシステムで相互に情報交換を行う必要がある．運転支援システムでは，センサデータから立体物の情報を生成する処理など，各アプリケーションで同一センサデータに対して共通する処理を行っており，センサやアプリケーションの増加に伴い，データ管理に伴うシステム開発コストの増大が課題となっている．

そこで様々なセンサのデータ処理（車載データ処理）をアプリケーションから切り離し，車載データとして統合的に管理するシステムが検討されている．欧州の SAFESPOT プロジェクトでは，他車両や交通情報など，車載データの内，車両の周辺情報を管理するプラットフォームとして LDM（Local Dynamic Map）が提案されている．LDM では，更新頻度の観点から 4 階層に分離して車載データを管理している．LDM では，車載データをリレーショナルデータベース管理システム RDBMS（Relational Database Management System）に格納する実装が示されており，また車載データにアクセスするための API を，RDBMS への問い合わせ言語である SQL（Structured Query Language）により実現する．しかし車載システムでは，動的に更新されるデータに対して 10～100ms 程の短い周期で処理し続ける，逐次処理が求められるため，蓄積された車載データに対して問い合わせることを前提とした LDM を適用するのは困難であった．

そこで車載データをストリームとして管理し，そのストリームの処理を，逐次処理可能なオペレータで定義する SPD（Stream Processing Description）と，その SPD を実行するデータストリーム管理システム DSMS（Data Stream Management System）を活用した，車載データストリーム管理システムが検討されている．SPD では，RDBMS とは異なり，車載データをストリームとして扱い，車載データ処理を，逐次処理可能なオペレータをデータフローで接続したクエリとして記述する．SPD により時々刻々とデータが更新される車載データの処理の保守性，再利用性を向上し，開発コストを削減することが期待される．

DSMS は，STREAM，Aurora，Borealis など，様々なプロトタイプ，製品が開発されているが，これらの DSMS は，株取引やパケット，ログ監視などを扱う汎用システムを想定しており，リソース使用量やリアルタイム性の観点から車載システムにそのまま適用することは難しい．ストリーム LDM など先行研究では，SPD の車載データ処理への適用性に関する検討が行われていたが，SPD のクエリをサーバ上で DSMS を実行することにより，クエリが実行可能であることを確認していた．車載システムは，汎用システムとは異なりハードウェアや OS が多種多様なバリエーションを持つため，多様な車載プラットフォーム上で動作させる必要がある（課題 1）．

車載システムではコストや信頼性の観点から，小容量の ROM，RAM や，低周波数の CPU が搭載される．しかし従来の DSMS で対象としているシステムは，その特性から，動的に実行するクエリが変化し，将来実行するクエリも予測できないため，様々なクエリをサポートする必要があり，実行環境に多くのモジュールを組み込む必要があり，ROM 使用量を削減する必要がある（課題 2）．また動的に変わるクエリや実行状況に従って処理最適化するため，ストリームキューで接続された各オペレータを，スケジューラが任意のタイミングや順序で呼び出し実行する．従ってストリームキューによる RAM 使用量の増加を抑え（課題 3），またスケジューリングの処理に要する CPU 処理時間も削減する必要がある（課題 4）．

また車載システムではプラットフォームにリアルタイム OS RTOS（Real-Time Operating System）を搭載し，全てのアプリケーションを RTOS のタスクに割り当て，タスクに優先度に従って実行を制御することでリアルタイム性を確保する．しかし DSMS では SPD のクエリに対して独自のスケジューリングを行い，RTOS を活用していない．そのため RTOS と連携し，SPD のクエリとその他のアプリケーションを統一的にスケジューリングすることで，リアルタイム処理を実現する必要がある（課題 5）．

そこで本研究では，課題 1～5 を解決するため，以下の 3 つの研究を実施する．

1 つ目の研究として，車載システムに適用可能な DSMS として，eDSMS（embeded automotive DSMS）を提案した．車載システムはアプリケーションが固定され，クエリにより記述された処理が動的に変更されないことに着目して，静的（設計時）に SPD のクエリを登録し，ランタイムジェネレータによりクエリから DSMS のランタイムコードを生成する．そしてクエリに従って必要最小限のモジュールのみを含むランタイムを生成することで，ROM 使用量を削減する．また異なるハードウェアや OS に対応したモジュールを持ち，実行環境のコンフィギュレーションに応じて適切なモジュールを選択することで，車載システムにおける様々なハードウェア/OS に適用する．eDSMS のプロトタイプを開発し，車載システムを想定した環境で評価した結果，コンフィギュレーションに応じて制御系および HMI（Human Interface）系のシステムを想定した環境のハードウェアや OS に適用することができた（課題 1 の解決）．また ROM 使用量を，従来の DSMS と比較し 46.6%削減し 102KB となり，車載システムで想定される ROM（384～1024KB）より小さくなることを確認した（課題 2 の解決）．

2つ目の研究として,eDSMSのストリームキューやスケジューリング処理を減らすことにより,RAM使用量及びCPU処理時間を削減することを目的とし,クエリや,そのオペレータの優先度に従って,静的にオペレータの動的な接続切替えを削減する方式（以下,静的スケジューリング方式）を提案した.静的スケジューリング方式では,車載システムにおいてクエリやオペレータの優先度が静的に決まることに着目し,優先度に従って複数のオペレータをグループとして抽出し,同一グループ内のオペレータ間のストリームキューを除去し,同期してオペレータを実行する.またグループ内において,動的にオペレータが切り替わらない箇所をクエリから検出し,検出箇所では,スケジューラを介さずオペレータが直接,次のオペレータを実行する.車載システムを想定した環境で静的スケジューリング方式を評価した結果を,従来のDSMSと比較しRAM使用量を54.7%削減し33KBとなり,車載システムで想定されるRAM（24～80KB）に搭載できる見込みであることを確認した（課題3の解決）.またCPU処理時間が従来のDSMSと比較し36%削減し670μ秒となった.車載システムにおける動作周期（10ms以上）と比較して十分小さいことから,車載システムのCPUにより処理可能である見込みを得た（課題4の解決）.

3つ目の研究として,性能要件を満たしつつ,リアルタイム性を確保する,クエリ処理最適化手法としてクエリコンテキスト共有方式を提案した.クエリコンテキスト共有方式では,アプリケーション間での優先度の逆転を防ぐため,クエリを,その処理結果を利用するアプリケーションと同一タスクに割り当て,同一優先度で実行する.そして処理が等価なクエリ間で,処理ではなく,クエリの処理過程で必要となるデータ（コンテキスト）を共有し,あるクエリの実行がスケジューリングの関係上,中断され,他のクエリが実行された場合,前のクエリのコンテキストを引き継ぐ.これによりクエリ処理最適化を実現する.評価の結果,クエリコンテキスト共有方式では,クエリ処理最適化によりRAM使用量,CPU処理時間を各々46%,32%削減しつつ,優先度逆転時間を100μ秒以下に抑えた.車載システムの動作周期は10ms以上であり,動作周期と比較し優先度逆転時間が十分小さいことから,リアルタイム性を確保できる見込みを得た（課題5の解決）.

以上より,本研究で提案した車載システム向けDSMSのランタイムコード生成技術とその最適化手法が,従来の汎用システム向けのDSMSにおける課題1～5に対して有効であることを,実用性の高いプロトタイプにより示した.本手法の適用により車載システム上で,SPDで定義したデータ処理処理を実行することが可能とし,運転支援システムにおけるデータ管理の開発コスト低減に大きく貢献することができた.