様式 C-19

科学研究費補助金研究成果報告書

平成 22 年　5 月　1 日現在

研究種目：　若手研究(B)
研究期間：　2008　～　2009
課題番号：　20700058
研究課題名（和文）　異種デバイス間連携のためのソフトウェア自動分散化手法
研究課題名（英文）　An Automatic Software Decentralization Method for Heterogeneous Device Collaboration

研究代表者
岩崎　陽平（YOHEI IWASAKI）
名古屋大学・大学院工学研究科・研究員
研究者番号：20447832

研究成果の概要（和文）：
単一のプログラムとして記述された集中制御型ソフトウェアを，複数の端末で動作する分散協調型ソフトウェアに変換する自動分散化手法を確立した．本手法では，入力プログラムの各命令文を実行に適した端末に分配し，端末間のメッセージ通信処理を生成する．この際に，プログラムの処理やデータの流れを解析することにより送信すべきメッセージの内容を抽出する．本手法をnesC言語に対して実装し動作検証と評価を行った．また，異種プラットフォームのソフトウェア開発の容易化についても検討を行った．

研究成果の概要（英文）：
The automatic software decentralization method for converting the centralized software described as a single program into the decentralized software that worked with two or more terminals was established. Each statement of the input program is distributed to the terminal that is appropriate for execution, and the message communication codes are generated. The contents of the message are analyzed by data flow analysis. We have implemented and evaluated the prototype system of the proposed method for the nesC programming language. We also examined the simplification of software development of different platforms.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2008 年度	1,700,000	510,000	2,210,000
2009 年度	1,600,000	480,000	2,080,000
年度			
年度			
年度			
総　計	3,300,000	990,000	4,290,000

研究分野：総合領域
科研費の分科・細目：情報学 ・ 計算機システム・ネットワーク
キーワード：
ユビキタスコンピューティング，分散協調ソフトウェア，センサノード，プログラミング，TinyOS，nesC，ソフトウェア自動分散化

１．研究開始当初の背景
近年， センサネットワーク用端末である MOTE や， 無線搭載のワンチップマイコンである nRF24E1（Nordic 社）など，無線通信機能を搭載した低コストな小型端末が開発されつつあり，センサ・家電・文房具などの様々な小型機器がネットワークに参加しつつある．これらの機器がネットワーク経由で連携して高度なアプリケーションサービスを実現し，我々の生活を支援することが期待される．
機能の異なる複数の機器（異種デバイス）を，ネットワーク経由で連携させるためのソフトウェア（連携ソフトウェア）は，主に集中制御型と分散協調型に分類される．集中制御型では，単一の連携ソフトウェアが，ネットワーク上の機器を遠隔制御して連携を実現する．一方，分散協調型では，各機器が特定の連携方法に特化した連携ソフトウェアを持ち，機器同士が直接通信して連携を実現する．分散協調型は，制御用端末を別途必要としない，特定の連携方法に特化した通信効率の良いメッセージ通信方式を使える，などの利点がある一方，連携ソフトウェアが複数端末に分散し，その開発や保守は容易ではない．

２．研究の目的
本研究では，異種デバイス間の連携のために，単一のプログラムで記述された集中制御型ソフトウェアを，自動的に分割し分散協調型ソフトウェアに変換する，自動分散化手法を提案する．特に，センサネットワーク用端末（センサノード）や産業用ワンチップマイコンなど，CPU パワーやメモリが制限された低レベルデバイス（CPU 8 ビット，メモリ数 KB 程度）で動作するソフトウェアを，分散化の対象とする．

３．研究の方法
以下の通り研究を実施した．

(1) ソフトウェアの自動分散化手法の確立
ソフトウェアの分散化は，以下のような手順で行う．まず，単一のプログラムとして記述された連携ソフトウェアを，命令文（ステートメント）単位で解析し，各命令文を実行に適した端末に分配する．次に，プログラムの処理やデータの流れを解析することにより，端末間のメッセージ通信処理を生成する．以上の処理を自動的に行うアルゴリズムを確立する．

(2)プロトタイプシステムの実装と評価
低レベルデバイス向けのプログラミング言語である nesC を対象として，提案した手法に基づくプロトタイプシステムを実装する．また，これを用いて，実際に TinyOS エミュレータや低レベルデバイス上で動作するソフトウェアを生成し，本プロトタイプシステムを評価する．

(3)異種プラットフォームへの適応
発展的課題として，プラットフォームや開発言語の異なるデバイスのソフトウェア開発の容易化についても検討を行う．

４．研究成果
本研究の成果を以下に示す．

(1)自動分散化アルゴリズム
単一のプログラムで記述された集中制御型のソフトウェアを，自動的に分割し分散協調型のソフトウェアに変換する，自動分散化手法のアルゴリズムを確立した．
本アルゴリズムでは，まず，入力されたソースプログラムの各ステートメント（命令）ごとに，それが実行される端末を決定する．そして，次に実行するステートメントが遠隔で実行するべきものの場合，メッセージ通信によりこれを実現する．このメッセージ通信に含まれる引数は，データフロー解析により決定される．以下に本アルゴリズムの概要をまとめる．

①ソースプログラムの入力
本アルゴリズムでは，単一の端末で動作するスタンドアロンのソフトウェア（集中制御型のソフトウェア）を，ソースプログラムとして入力する．本アルゴリズムでは，端末固有の機能の利用は，関数呼び出しの形で記述する．このため，各関数をどの端末で実行するかを記述した，機能配置情報を入力する．

②制御フローグラフの作成
次に，入力されたソースプログラムから，制御フローグラフを作成する．制御フローグラフとは，プログラムを複数のブロックに分け，ブロック間の制御の流れ（制御フロー）を矢印で表したものである．ブロックには，複数のステートメント（文）が含まれる．本アルゴリズムでは，各ステートメントをそれぞれいずれかの端末へと配置するため，複数の関数呼び出しを含むステートメントは，本ステップにおいて，一時変数を使って複数のステートメントへと分割する．

③各ステートメントを実行する端末の決定
次に，各ステートメントを実行する端末を決定する（ステートメントの配置）．端末固有の機能（機能配置情報で指定された関数）を呼び出しているステートメントは，その機能が存在する端末上で実行される．それ以外の（特定の端末に関連づけられていない）ス

テートメントの配置は任意であり，様々な方法が考えられる．例えば現在の実装では，プログラム上で直前にあるステートメントと同一の端末上で実行する．

④実行端末ごとにブロックを分割
次に，ステートメントを実行する端末が切り替わる点で，ブロックを分割し，それらのブロック間を矢印でつなぐ．その結果，各ブロックは単一の端末で実行されるステートメントのみを含むようになる．その端末をそのブロックの実行端末と呼ぶ．

⑤データフロー解析
次に，ブロック間で受け渡すべきデータを判断するために，データフロー解析を行う．本アルゴリズムでは，ブロック間で受け渡すべきデータとして，生きている変数を用いる．生きている変数とは，ブロックの入口での値が（そのブロックまたは後続のブロックで）使われる可能性のある変数のことである．

⑥メッセージ送信コードの生成
本アルゴリズムでは，遠隔の端末への遷移（実行端末が異なるブロック間のデータフロー）を，メッセージの送信として実現する．遷移先のブロックごとにメッセージの種類が作られ，変数の値を同期するために，遷移先のブロックの入口で生きている変数をメッセージの引数として渡す．

(2)プロトタイプシステムの実装と評価
低レベルデバイス向けのプログラミング言語である nesC を，入力および生成される対象のプログラム言語として，提案した手法に基づくプロトタイプシステムを実装した．nesC は C 言語を拡張したプログラミング言語であり，MOTE 等の低レベルデバイスを対象とし，静的な最適化を適用できるコンポーネントモデルを採用している．なお，自動分散化のアルゴリズム自体は低レベルデバイス上で動作させることは想定していないため，本アルゴリズムは JavaSE 6，および nesC のパーサである TinyDT を用いて実装した．
本プロトタイプシステムでは，端末固有の機能を，nesC のインタフェースとして表し，また，各インタフェースがどの端末上に存在するかを機能配置情報として入力する．また，分散化させたいソースプログラムは nesC のタスク関数として記述する．
まず，本プロトタイプシステムにより自動生成されたプログラムを，複数の PC 上で TinyOS エミュレータである EmTOS を用いて動作させ，動作確認を行った．動作の様子を図 1 (1)および図 1(2)に示す．自動分散化の対象としたアプリケーションは，センサ端末上のセンサにより気温とバッテリー電圧を測定し，LCD 端末上に，測定された温度値と，バッテリー電圧が閾値より小さい場合には警告メッセージを表示する．また，開発の容易さの簡単な評価として，プログラムのサイズを測定した．表 1 に，入力したプログラム，および生成された分散ソフトウェアのプログラムのサイズを示す．

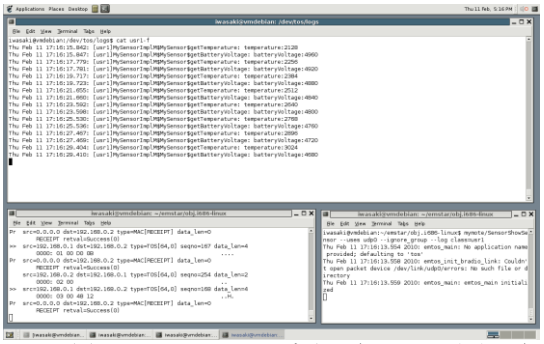
図 1 (1)．EmTOS 上での実行（センサ端末側）

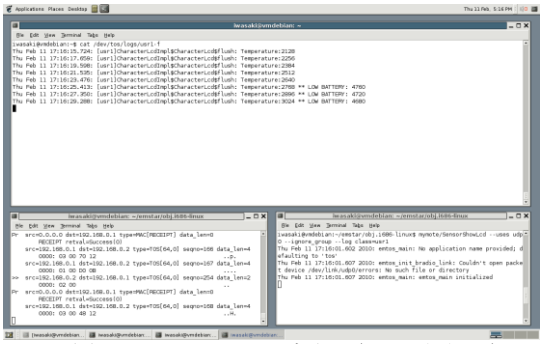
図 1 (2)．EmTOS 上での実行（LCD 端末側）

表 1．センサ値表示プログラムのサイズ

	行数	文字数
入力したプログラム	58 行	1744 文字
生成されたプログラム	207 行	6265 文字

次に，実際の低レベルデバイスである nRF24E1（Nordic 社）上での動作確認も行った．動作の様子を図 2 に示す．自動分散化の対象としたのはモーニングコールサービスのアプリケーションであり，センサ端末上で明るさを測定し，明るさが閾値より大きい場合に，LCD 端末上にモーニングコールのメッセージを表示し，スピーカ端末上でアラーム音を発生する．先ほどと同様に，表 2 に，入力したプログラム，および生成された分散ソフトウェアのプログラムのサイズを示す．
表 1 および表 2 の結果より，分散ソフトウェアの実装には大きなサイズのプログラムが必要であり，提案手法によって，実際に書かなければならないプログラムのサイズを減少できたことが分かる．

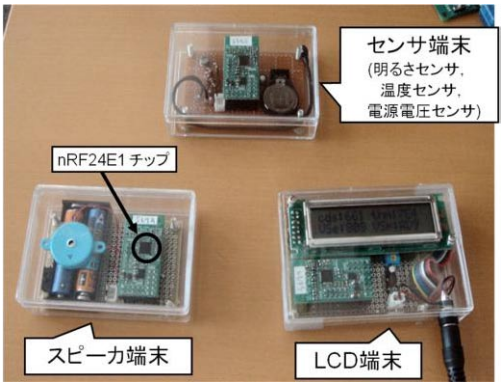

図 2. nRF24E1 チップ上での実行

表 2. モーニングコールプログラムのサイズ

	行数	文字数
入力したプログラム	55 行	1553 文字
生成されたプログラム	215 行	5975 文字

(3) 異種プラットフォームへの適用
以上により，nesC 言語上での自動分散化手法が実現できた．一方，単一のプログラムを記述するだけで，TinyOS（nesC 言語の動作環境）だけでなく，他の様々なプラットフォーム上で動作する分散ソフトウェアが開発できれば便利である（クロスプラットフォーム開発）．そこで発展的課題として， TinyOS 以外のプラットフォームに対するソフトウェア開発の容易化についても検討を行った．例えば，Java 言語でソフトウェアを開発できるセンサノードである Sentilla ノード（加速度センサを搭載）や，近年普及しつつある iPhone，Android 端末などのスマートフォン（加速度センサ，地磁気センサ，マイク，GPS 等を搭載）などのプラットフォームが挙げられる．これらの様々なプラットフォーム間の差異を吸収し，クロスプラットフォーム開発を行うためのアプローチを検討した．

①Web アプリケーションとして実装する方法
まずは，アプリケーションを Web ブラウザ上で動作する Web アプリケーションとして実装する方法を検討した．標準化された HTTP，HTML5，CSS，JavaScript などの技術仕様に基づいてアプリケーションを記述することにより，異なるプラットフォーム上で動作する Web ブラウザ上で，アプリケーションを動作させることができる．アプリケーション例として，まず，REST に基づく Web サービスとして公開されたスマートルーム内の機器を，スマートフォン上で操作することができる REST リモコンアプリケーションを作成した．次に，カメラを搭載したキオスク端末上に表示された認証コード（4 桁の数値）を，スマートフォン上で入力することにより，写真の撮影が行えるキオスクカメラアプリケーションを作成した．本アプリケーションが iPod touch（iPhons OS 3.1）と Android 端末（Android OS 2.1）上で動作している様子を図 3 に示す．
ただし，Web ブラウザを実装するにはある程度のリソースが必要であり，低レベルデバイス上での動作は難しい．また，JavaScript 上からセンサの値を取得する API がまだ十分に標準化されていない，などの課題がある．

図 3　異なるプラットフォームのスマートフォン端末上での実行

②ラッパーライブラリとトランスレータを用いる方法
次に，API の差異を吸収したラッパーライブラリを開発し，異なるプログラミング言語間のトランスレータを用いる方法を検討した．まずは，各プラットフォームの開発環境や API の差異を調査した．一般的なワンチップマイコン（例えば nRF24E1，8051，H8 など）では，C 言語による開発を行うことが多い．本研究で対象とした Tiny OS では，C 言語を拡張した nesC 言語を用いて開発を行う．nesC コンパイラは C 言語へのトランスレータとして実現されており，対象となるデバイス自体は C 言語による開発も可能である．Sentilla ノード（Sentilla Corporation）は，Java による開発が可能な低レベルデバイスである． iPhone OS（Apple）は， Apple 社製端末用の OS であり，Objective-C，C 言語，C++などの言語による開発が可能である．Android OS（Google）は，スマートフォン向けの OS であり，Java による開発が可能である．iPhone と Android の API を比較すると，開発言語の差異の他に，iPhone OS では（現時点では）バックグランドサービスを動かせない，Android ではユーザインタフェースの画面（Activity）間でオブジェクトを直接共有出来ないなど，プロセスモデルの差異も多い．
開発言語間のトランスレータは，既に多くのものが開発されている．例えば，C 言語から Java へのトランスレータは，Stonis らの C to Java converter，SoftLogica 製

の C To Java Converter，Novosoft 製の C2J Converter などがある．これらのトランスレータを用いれば，例えば C 言語で書かれたワンチップマイコン向けのプログラムを，Java を用いる Android OS 向けソフトウェアにリンクすることができるだろう．ただしこれらの異種プラットフォーム上で共通のプログラムを動作させるためには，プラットフォーム間の API の差異を吸収するラッパーライブラリを開発する必要があり，これは今後の課題である．

(4)提案手法のインパクト

提案した自動分散化手法を用いれば，開発時には集中制御型の利点（連携ソフトウェアを単一のプログラムで容易に開発可能）を得られ，実行時には分散協調型の利点（通信効率の良いメッセージ通信方式で直接通信可能）を得られるため，連携ソフトウェアの開発と動作の効率が向上する．

RPC（Remote Procedure Call）を用いた従来の自動分散化手法では，遠隔の機能を使う度に往復の通信を行う．一方提案手法では，プログラムの制御の流れの中で，機能を実行すべき端末が切り替わる地点でのみ通信を行う．このため，同一の端末の機能を複数回連続して利用する際に，それらを通信無しにまとめて実行することができ，通信回数を削減できる可能性がある．また，RPC の利用には一般にマルチスレッドが必要となるが，提案手法では，プログラムを複数の関数（ブロック）に分割することにより，シングルスレットでの動作を可能としている．このため，生成された分散ソフトウェアが，低レベルデバイス（CPU8bit，メモリ 4KB 程度のワンチップマイコン等）の上でも動作する．低レベルデバイスは，多数の無線小型端末が協調して環境情報を収集するセンサネットワークの分野や，日常生活の至る所にコンピュータが組み込まれるユビキタスコンピューティングの分野で幅広く利用されており，提案手法の応用分野は広い．

(5)今後の展望

今後の課題としては，以下のようなものが挙げられる．

①より大規模なアプリケーションへ提案手法を適用し，本手法の有用性を示す．

②入力するソースプログラムの仕様を拡張し，同類のセンサノードをまとめて扱ったり，通信エラーなどの例外処理を記述可能にする．

③自動的に生成されるメッセージ通信処理を最適化する

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔学会発表〕（計 4 件）

① 北出 卓矢，岩崎 陽平，河口 信夫，平野 靖，梶田 将司，間瀬 健二，“プライバシー保護を考慮した公共センサ応用，” 電子情報通信学会総合大会，2010 年 3 月 18 日，仙台（東北大学）

② Takuya Kitade, Kosuke Niwa, Yuichi Koyama, Kazuhiro Naito, Yohei Iwasaki, Nobuo Kawaguchi, Yasushi Hirano, Shoji Kajita, Kenji Mase, “A location-based application with public anonymized sensor data for personal use,” International Conference on Security Camera Network, Privacy Protection and Community Safety 2009 (SPC2009), 2009 年 10 月 29 日，桐生（桐生市民文化会館）

③ Nobuo Kawaguchi, Nobuhiko Nishio, Yohei Iwasaki, Ismail Arai, Koichi Tanaka, Shigeo Fujiwara, “Secure and Dynamic Coordination of Heterogeneous Smart Spaces”, UbiWORK: Design and Evaluation of Smart Environments in the Workplace (International Conference on Ubiquitous Computing, UbiComp2008 Workshops), 2008 年 9 月 21 日，ソウル（COEX）

④ 岩崎 陽平，榎堀 優，藤原 茂雄，田中 宏一，西尾 信彦，河口 信夫，“REST に基づく異種スマート環境間のセキュアな連携基盤，” マルチメディア，分散，協調とモバイル(DICOMO2008)シンポジウム論文集，2008 年 7 月 8 日，札幌（定山渓ビューホテル）

６．研究組織

(1)研究代表者

岩崎　陽平（YOHEI IWASAKI）

名古屋大学・大学院工学研究科・研究員

研究者番号：20447832