

携帯電話アプリケーションによるモバイルな センサネットワーク構築支援システム

Wireless Sensor Network Constructing System for a Mobile Phone

大林真人 † 西山裕之 ‡ 溝口文雄 ‡

Makoto OBAYASHI, Hiroyuki NISHIYAMA and Fumio MIZOGUCHI

† 東京都立産業技術研究センター IT グループ

IT Group, Tokyo Metropolitan Industrial Technology Research Institute

‡ 東京理科大学 理工学部

Faculty of Science and Technology, Tokyo University of Science

obayashi.makoto@iri-tokyo.jp, nisiyama@ia.noda.tus.ac.jp, mizo@ia.noda.tus.ac.jp

携帯電話アプリケーションによって、センサノードの動作構築環境を提供するシステムを開発する。近年、様々な種類の多数のセンサノードを屋外に配置することによって広域的な環境情報をセンシングし、安全・安心を実現するユビキタスコンピューティング環境の構築が実証実験を通じて注目を浴びている。しかしながら、一般ユーザの利用方法は、端末を経由してシステムにアクセスするに留まり、web のようにコンテンツの提供やシステム構築に対して自由に参加することは出来ない。本研究による成果物は、一般的な携帯電話を用いて、「いつでも、どこでも」センサネットワークを用いたユビキタスな環境を構築する簡易な手段を提供するものである。

1 はじめに

ワイヤレスセンサネットワークを構築するセンサノードデバイスには、既に多くの種類が存在する。特に、小型かつ低消費電力性を追求して設計されたセンサノードは、実環境における様々な場所や物体に容易に設置することができるだけでなく、内蔵されたバッテリーによって年単位での長時間駆動も可能となる [1]。更に、様々なセンサやアクチュエータを組み合わせることによって、より多機能なセンサネットワークおよびロボティックルームを簡便に構築することも可能となると思われる [3]。これにより、センサネットワークは非常に幅広い応用性を持ち、産業用途だけでなく、一般向けの多くのニーズを内包した技術と考えられる。特に、総務省による u-Japan の政策に見られるように、高齢化社会および犯罪の多発を背景とした「安全・安心」を屋内外で多様かつ簡易に実現するデバイスとして、多くの普及が望まれる。しかしながら、センサノードは、設置する対象や場所、用途や接続するセンサの種類にしたがって、その動作を変更する必要がある、その設定には、プログラミングに関する知識と技術が必要とされる。よって、一般のユーザがワイヤレスセンサネットワークのシステム構築に参加し、多様な用途に適用させ

ることは、技術的に困難な作業となる。そのため、センサネットワーク技術が十分な汎用性を持っているにも関わらず、エンドユーザによって要求される機能が提供されずに、限られたエンジニアによる限定された分野および事例への適用のみに留まる恐れがある。また、屋外で使用するためには、設置環境や現場における容易な開発、動作設定が行えることが望ましい。これらを考慮すると、ワイヤレスセンサネットワークを、より身近なデバイスとして普及させるためには、携帯可能かつ汎用的な情報端末と直観的なインターフェースを採用した開発環境の提供が望ましいと考えられる。

本研究では、上述した問題を解決するために、携帯電話アプリケーションによって、センサノードの簡易的な開発環境を提供するシステムを開発する。また、対象とするワイヤレスセンサネットワークとして、設置の簡易性・可搬性に優れた小型かつ低消費電力のデバイスをターゲットとする。これは、GUI ベースによる開発環境を提供し、「いつでも、どこでも」センサノードのプログラミングを提供することを可能とするものである。現在、我が国における携帯電話の普及率は非常に高いだけでなく、通信速度の向上、機能の高度化が進み、インターネットへの接続数は通常の計算機によるものを超えるまでに至っ

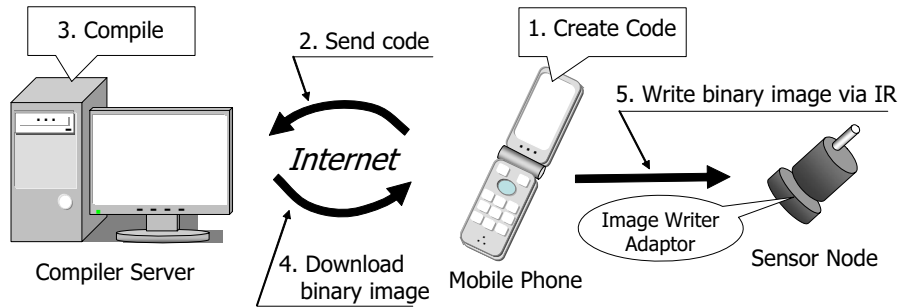


図 1: System Configuration of our Mobile WSN Developing Environment

ている [5]。これらの事実を考慮すると、携帯電話こそが、センサネットワークの簡易開発環境の動作プラットフォームとして十分に汎用的かつ実装可能なデバイスであると思われる。しかしながら、携帯電話と通常の計算機とを比較した場合、ハードウェアと OS の性能のみならず、表示および操作インターフェースにおいて大きな制約が存在する。よって、我々は、携帯電話上の限定された描画空間 (240×240pixel) に適合した GUI と視覚化表現によるセンサネットワークの開発環境を提案する。また、GUI によって生成されたセンサノードの動作定義をネイティブコードに変換する手法を示す。これにより、一般ユーザによるセンサネットワークデバイスを用いたコピキタス環境の開発を容易にする。

2 設計方針

2.1 携帯電話による構築システム

図 1 に本研究におけるシステム構成を示す。一般的に、携帯電話上では、メモリ空間および CPU の能力から、高い負荷を必要とするアプリケーションを動作させることには限界がある。このため、本システムでは、携帯電話網からのインターネット接続を利用して、特定のサーバに接続させ、ユーザによって構築された動作定義ファイルのコンパイル処理を実現する。本システムのユーザは、所有する携帯電話に実装された専用アプリケーションソフトウェアを使用することによって、センサノードの動作を定義する。携帯電話上で構築された動作定義ファイルは、ネットワークを介してコンパイラサーバに送信され、センサノード上で実行可能なバイナリイメージが生成される。バイナリイメージは、コンパイラサーバから送信され、携帯電話上のメモリに格納される。ユーザは、携帯電話に装備された赤外線ポートを使用して、センサノードにバイナリイメージの

書き込みを行う。ここで、通常の計算機におけるプログラミングと比較すると、表示および操作インターフェースの点で大きな制約を受けることとなる。次に、限定されたインタフェース環境におけるセンサネットワークおよびセンサノードの動作定義システムとその視覚化技法についての詳細を述べる。

2.2 GUI ベースによる動作定義手法と視覚化技術

携帯電話が一般の計算機や PDA と比較して劣る点は、描画領域が狭く、操作インターフェースが貧弱であることであることが挙げられる。このため、携帯電話上でプログラミングのようなコード記述作業を行うことは、大きな苦痛を伴う労働となりうる。よって、単純な操作によってセンサノードの動作を定義するためには、GUI をベースとした開発環境を使用することが、一つの解であると考えられる。ここで、本研究における動作定義 GUI は、以下の特徴を備えることを重点に設計を行った。すなわち、(1) 携帯電話の限定された操作インターフェースによる十分な操作性の実装、(2) 限定された描画エリア内における効果的な視覚化、の 2 点である。本研究では、各センサノードの振舞いを GUI で定義するため、その動作を UML2.0 に準じたステートマシン図によって表現した。ステートマシン図とは、あるオブジェクトが、時間の経過やイベントの発生と取得によって、その状態どのように変化するかを表現するダイアグラムである。ステートマシン図は、状態ノードと呼ばれる複数の形状を持つ記号と、遷移の方向を示す矢印の集合によって構成される。このとき、状態ノードの数が増加すると、携帯電話の限られた描画領域では表示することが困難となる。このため、ステートマシン図の全体を効率的にユーザに伝達するために、3D による視覚化アルゴリズムを開発した。この手法は、ステートマシン図の全体を一画面内に表

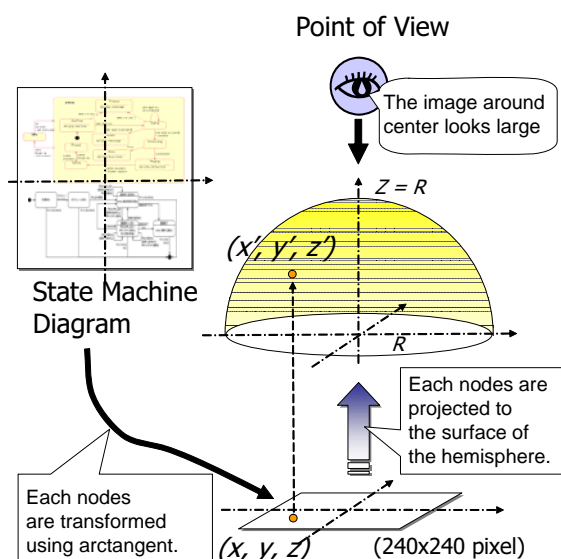


図 2: Information Visualization Technique in Our System

示する一方で、ユーザの注視領域を拡大して表示する(図2)。GUI上における各状態ノードの描画変換位置は以下のアルゴリズムによって算出される。ここで、平面内に配置された、ある状態ノードの座標を (x_{pi}, y_{pi}, z_{pi}) ($i = 1 \dots n$) とすると、3次元座標 (x_{si}, y_{si}, z_{si}) に投影される変換式は以下のように表される。

$$\begin{aligned} (x_{si}, y_{si})^T &= C_s \tan^{-1} Z_s (x_{pi}, y_{pi})^T \\ z_{si}^2 &= R - (x_{si}^2 + y_{si}^2), \quad (z_{si} \geq 0) \end{aligned}$$

ここで、 Z_s , C_s , R は収束係数、描画領域係数、曲率係数であり、これらの係数を変更することによって、3次元に投影されたステートマシン図のバランス、注視領域の拡大率を変更することが可能である。図3に、我々の視覚化手法による描画結果を示す。図より、全ての状態遷移ノードを一画面内に描画しつつ、ユーザの注視領域となる中央付近に位置するノードは拡大されて表示されることが確認できる。逆に、その他の状態遷移ノードは、画面端に縮小されて表示されることが確認できる。各状態ノードの動作内容は、編集フォーム上で、システムコールやイベント種別等を選択することによって実現される。

2.3 センサノード動作処理系

GUIによって定義されたセンサノードの振舞いは、アプリケーション内部でプログラミング言語に変換する必要が生じる。ここで、ステートマシン図におけ

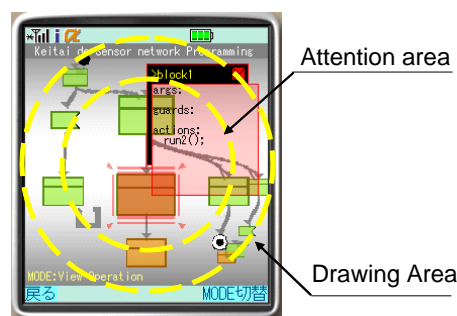


図 3: The Snapshot of Development Environment in Mobile Phone

る表現要素である、各種状態や遷移、遷移条件、送受信シグナル等の記述をネイティブコードに一意に変換することが必要となる。UMLによる記述を拡張することによって自律したソフトウェアプロセスの振舞いの定義と実装を可能とする手法は既に提案されているが[2]、限定されたりリソースを持つセンサノードに対して従来の手法を用いることは困難となる。

本システムにおいては、我々の先行研究によって開発した言語および言語処理系である、TinyMRL[4]を使用することによって、GUIによる動作定義とコード生成のシームレスな変換を実現した。TinyMRLはルールベースによって状態を記述することが可能であり、GHC(Guarded Horn Clause)による記述形式と同様に、ルール名、条件節、実行節の3部から構成される。そして、実行可能状態にあり、かつ条件節の内容とセンサノードの内部状態が一致したルールが実行される。この構造により、UMLによるステートマシン図との対応が容易であり、さらにGUIベースでの編集に適している。また、ファシリテータを使用した、“subscribe”、“broker”、“recruite”、“recommend”等の動的なサービス発見機能をシステムコールとして実装しており、アドレスや各ノードに設定された名称をベースにした静的なネットワークアクセスだけでなく、機能や場所などの条件を基にした動的なノード発見と通信を可能とし、各種透過性を実現する。これにより、センサネットワーク上のノード構成の変化や新たな機能の追加に対して柔軟なシステムを構築することが容易となる。

処理系内部において、各ルールは、名称および引数の数によって分類される。すなわち、同じルール名と同じ数の引数を持つ全てのルールは同一のグループに分類され、2つ以上の同じグループのルールが同時に呼び出されることは無い。呼び出されるルールは、条件節における定義式が、呼び出し元による引

数の値およびセンサノードの内部状態が一致するものだけが呼び出される。ここで、条件に適合するルールが複数存在する場合には、条件節内における定義式の数にしたがって優先度が決定され、最も多い条件を持つルールが最高優先度として処理される。最高優先度となるルールが複数存在する場合には、実行待ちキューの先頭に格納されたルールが選択される。また、各ルールの実行節の最後に自身のルールを再帰的に呼び出すことによって、その同じグループに属するルールの集合によって構成される一連の処理が継続される。同じグループに属するルールは単一の状態を表現しており、再帰的に自己を呼び出すことによって、同じ状態での動作を保持する。また、他のノードからの通信や各種のイベントにしたがって、自己を他の状態へ遷移させるときには、条件節にイベントを取得するためのシステムコールを定義し、実行節の最後に他のグループに属するルールの呼び出しを行うことによって実現される。状態の動作の終了は、実行節内部でのルール呼出を行わずに定義を完了することによって実現できる。このとき、どのルールも実行待ち状態にならないため、実行の終了と同時に一連の状態遷移動作は完全に終了する。

3 実装

DoJa-4.0 による i アプリ (携帯電話 901iS およびその上位機種対応) によって実装を行い、N901iS (NEC) 上で動作確認を行った。コンパイラサーバには、linux (Fedora core4) を使用しており、Tomcat+Servlet によってユーザからのリクエストが処理される。コンパイラサーバから携帯電話に送信されたセンサノードのバイナリイメージは、スクラッチパッドに保存される。よって、アプリケーションを終了しても端末上から消去されないため、コンパイル済みのバイナリイメージは何度でも使用することが可能である。携帯電話からセンサノードに対するバイナリイメージの書込みは、赤外線ポートによる IrOBEX プロトコルによって実現される。

また、本システムは、東京都立産業技術研究センターで開発されたセンサノードデバイス (図 4) に対応して実装された。CPU には H8S2238R (ルネサステクノロジ)、7.3728MHz を採用しており、256kbyte の ROM と 16kbyte の RAM 空間を持つ。センサノードの OS には、組込み OS であり、 μ ITron version4.0 仕様に準拠した、NORTi(株式会社ミスポ) を使用し

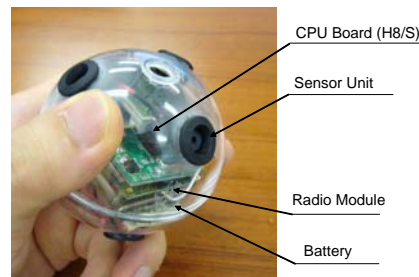


図 4: The Experimental Sensor Node

た。このセンサノードには、2.4GHz 帯の特定小電力による無線ユニットを使用しており、理想状態で約 300m の通信距離を持つ。

4 まとめ

本研究において、我々は一般の携帯電話上で動作するアプリケーションとして、各センサノードの振舞いを定義する開発環境を実現した。携帯電話は、通常の計算機環境と比較して、計算機リソースや描画領域、および操作インターフェースに大幅な制約を受けることとなる。このため、一部のタスクをネットワーク経由でサーバに処理させるだけでなく、携帯電話に特化した視覚化手法を用いることにより、GUI ベースでの開発環境を実現した。汎用的な携帯電話と、GUI による直観的な開発環境を提供することにより、ユビキタス環境構築ツールとしてのセンサネットワークの活用を拡大することが出来るものと考えられる。

参考文献

- [1] B.Warneke, M.Last, B.Liebowitz, and K.Pister. "Smart dust: Communicating with a cubic-millimeter computer," IEEE Computer, pages 44-51, January 2001
- [2] J.Odell, V.M.Parunak and B.Bauer: "Representing Agent Interaction Protocols in UML," in Proc. Agent Oriented Software Engineering, LNCS 1957, Springer-Verlag, 121-140, 2000.
- [3] M.Obayashi, H.Nishiyama and F.Mizoguchi: "Secure Cooperation in a Distributed Robot System using Active RFIDs," International Journal of Artificial Life and Robotics Vol.10, pp29-34, 2006.
- [4] M.Obayashi, H.Nishiyama and F.Mizoguchi, "Secured Cooperative Multi-Agent Framework in Limited Resources for Intelligent Sensor Network," Proceedings of the annual conference IECON, pp.2668-2673, Nov 2005.
- [5] 総務省, "平成 17 年「通信利用動向調査」の結果", 情報通信政策局総合政策課情報通信経済室, 19 May 2006.