社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－UBI－11 (9)
2006／5／23

無線センサノードを用いた日常生活サービス構築機構の提案

小泉健吾[1]，中井彦一郎[2]，榊原寛[1]，出内将夫[1]，高汐一紀[1]，徳田英幸[1,2]
{mics, hiko, skk, ide, kaz, hxt}@ht.sfc.keio.ac.jp
[1] 慶應義塾大学大学院 政策・メディア研究科
[2] 慶應義塾大学 環境情報学部

本研究では無線センサノードを用いた日常生活サービスの構築機構を提案する．近年，様々なユビキタスコンピューティング環境が構築されてきたが，手間やコストが問題となっていた．我々は，ユーザが普段利用にもかかわらず知的化が行われていない日用品 (モノ) に焦点を当て，無線センサノードを用いた低コストかつ即興的な知的化を目指した．本論文では，センサノードを用いたモノの知的化による日常生活サービス構築の際の課題について述べ，複数のセンサノードを用いて状況に応じたサービス提供を行うためのセンサノード用プラットフォーム「uElement」を設計・実装した．評価の結果，uElement がセンサノード上で実現可能であることを示した．

A Research on a Platform for Developing Daily Services with Wireless Sensor Nodes

Kengo Koizumi[1], Hikoichiro Nakai[2], Hiroshi Sakakibara[1], Masao Ideuchi[1], Kazunori Takashio[1], Hideyuki Tokuda[1,2]

[1] Graduate School of Media and Governance, Keio University
[2] Faculty of Environmental Information, Keio University

This paper proposes a platform for developing daily services with wireless sensor nodes. Recently, many ubiquitous computing environments have been realized. However, the costs of realizing ubiquitous computing environments still remain as problems to be solved. We focus on daily objects that users use everyday and are not smart, and approach to make them smart using wireless sensor nodes. This paper describes assignments for making daily objects smart using sensor nodes, and then describes our platform for sensor nodes, named "uElement", that enables a number of sensor nodes to provide services based on their conditions. Lastly, this paper proves our platform is able to work in a sensor node.

1 はじめに

近年，日常生活空間においてユビキタスコンピューティング環境 [13](以下，Ubicomp) を実現するため，様々な研究が行われている．Ubicomp では，実世界の情報に応じたサービスをユーザに提供可能となる．Ubicomp を日常生活空間で実現するためには，現在の家庭環境に普及しつつある PC やネットワークに加え，実世界の情報を取得するためのセンサやユーザに対しサービスを提供するアクチュエータが必要になる．予めセンサやアクチュエータが組み込まれていない空間や機器にセンシング能力やアクチュエーティング能力，及び計算能力を与えることを本論文では知的化と呼ぶ．

どのような対象にセンサやアクチュエータを組み込むかにより，知的化を三種類に大別できる．まず，Easy Living[6] や SSLab[11] のようにセンサやアクチュエータを部屋やオフィスに埋め込む「環境の知的化」，次に Smart Furniture[8] や Degital Decor[15] のようにセンサやアクチュエータを家具に組み込む「家具の知的化」，そして MediaCup[5], SPECs[10], Cooperative Artefacts[12] のようにモノへセンサやアクチュエータを組み込む「モノの知的化」の三種類である．

しかし，現状ではこれらの知的化を一般のユーザが行えないため，Ubicomp からサービスを十分に享受することは難しい．なぜなら，知的化を行う際に時間や費用面での莫大なコストが生じるため，既存研究における Ubicomp でのサービス利用モデルは，第三者が知的化した部屋，家具，モノをユーザが利用する形になる．このモデルではユーザの意思が知的化に反映されず，ユーザが独自に知的化したい環境，家具，モノがあったとしてもそれらを知的化することはできない．そのため，Ubicomp から提供されるサービスがユーザのニーズとかけ離れるという問題が生じる．

我々は，ユーザ自身によるモノの知的化に着目し，無線センサノードを用いたモノの知的化に焦点を当てた (図1)．無線センサノードとは，センシング能力，アクチュエーティング能力，計算能力を持った無線デバイスであり，代表的なセンサノードとして MICA2 DOT[1], Smart-Its Particle[7], uPart[4] などが挙げられる．モノに着目した理由として，モノはユーザの生活空間に存在するためユーザとのインタラクションが多く，モノを知的化することによって「落し物発見」や「盗難防止」などの日常生活上のサービスを構築し易い点が挙げられる．また，モノに取り付け可能無線センサノードによって，センサやアクチュエータをモノに組み込む必要がなくなるため，従来の手法よりも低コストで即興的にモノを知的化できる．さらに，モノにセンサノードを取り付けたり，必要のなくなったモノからセンサノードを取り外すといった，ユーザ自身のニーズに基づくモノの知的化が可能になる．

センサノードによるモノの知的化を実現するためには，モノにセンサノードを取り付けると共に，センサノードからユーザに対してサービス提供を行うための汎用的なプラッ

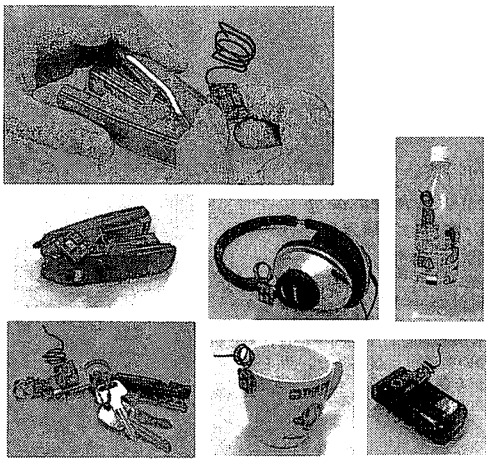

図 1: センサノードによるモノの知的化例

トフォームが必要不可欠である．既存の研究では，モノに対してセンサノードを固定的に取り付けていたが，本研究ではユーザによって動的にセンサノードが取り付けられるため，センサノードの取り付けからサービス提供までを支援するプラットフォームが重要である．このプラットフォームを実現するため，本研究ではまずセンサノードによるモノの知的化を行う際の課題について議論を行い，複数のセンサノードから状況に応じたサービス提供を可能にするセンサノード用プラットフォーム「uElement」を構築した．

本論文ではまず，第 2 節でモノを知的化することによって実現される世界について述べる．次に第 3 節でセンサノードを用いて知的化を実現するための課題について述べ，第 4 節で uElement の機能要件について述べる．第 5 節で uElement の設計を示し，第 6 節で今回作成したプロトタイプについて，第 7 節で評価について述べる．さらに，今後の課題について第 8 節で述べ，最後に結論を第 9 節でまとめる．

2 シナリオ

本節では，モノの知的化で実現されるシナリオについて述べる．まずモノの知的化によって提供されるサービスについて述べ，次にモノの知的化手順について述べる．

2.1 モノの知的化によるサービス例

モノを知的化することによって実現されるサービスは，日常生活を送る上で，「スケールは小さいが，あったら便利」なサービスでる．具体的なサービス例として，盗難検知サービス，落し物検知サービス，探し物発見サービスなどが挙げられる．以下に各サービスの詳細を示す．

盗難検知

携帯電話，手帳，財布などのユーザにとって大事なモノを知的化すれば，盗難検知サービスを実現できる．例えば，カフェや図書館などでは，携帯電話や手帳が入った鞄を机に置いたまま，注文や本を探すために席を外すことがある．しかし，悪意のある人によって携帯電話や手帳が鞄ごと盗まれてしまう可能性がある．この時，携帯電話や手帳を知的化しておけば，ユーザが近くにいないにも関わらず携帯電話や手帳が動かされた場合に，携帯電話や手帳から警告音を発生させて盗難を防げる．

落し物検知

盗難検知と同様に，ユーザにとって大事なモノとユーザが常に身につけているモノを知的化すれば，落し物検知サービスも実現できる．例えば，外出中に鞄から財布が落ちた場合，ユーザの身につけている時計やベルトが，財布が離れたことを検知し，盗難防止と同様に警告音を発してユーザに落し物の存在を知らせられる．また，ユーザが身につけているモノを利用しなくても，モノ同士が互いの動きを共有し合うことによって，ひとつだけ動きが止まったモノを落し物として検知することも可能である．

探し物発見

よく置き場所を忘れてしまうモノにセンサノードを取り付ければ，探し物発見サービスを実現できる．例えば，あるユーザはコンタクトレンズを外す時，メガネの入ったメガネケースが見つからず，探し回ってしまうことが多いとする．もしコンタクトレンズのケースとメガネケースが知的化されて互いに通信し合っていれば，一方を動かしたり強く振ったりすることでもう一方を鳴らしたり光らせたりし，ユーザに置き場所を知らせられる．また，ホームサーバに検索アプリケーションを置いておくことで，携帯電話や PC を使って実世界上のモノを検索することも可能になる．

2.2 センサノードによる知的化手順

本研究では知的化の道具としてセンサノードを用いた．以下に，センサノードを購入してからサービスが提供され始めるまでの手順を示す．

(1) センサノードを購入する
(2) モノに取り付ける
(3) サービスの準備を行う
(4) サービスを稼動させる

まずユーザはセンサノードや複数のサービスインストーラが入ったパッケージを購入し，センサノードをモノに取り付ける．センサノードを取り付ける対象は，ユーザが自身の興味に基づいて決定できる．次に，サービスのインストーラを利用し，ユーザが利用したいサービスを PC やセンサノードなどにインストールする．この時，ユーザはパッケージに含まれるエディタを利用し，パッケージに含まれるサービスだけでなく，自身のニーズに応じたサービスを構築できる．最後に，実際に PC やセンサノードなどにインストールしたサービスを利用する．

3 課題

本節では，まず 2 節で述べたシナリオに基づき，センサノードを用いたモノの知的化を実現するために解決すべき課題について述べる．次に，uElement が解決する課題について述べる．

3.1 センサノードによる知的化の課題

センサノードを用いたモノの知的化を実現するには，以下に述べる4つの課題があると考えられる．

1. **モノとセンサノードのアソシエーション**
   2.2項で知的化の手順について述べたが，モノにセンサノードを取り付けただけでは，どのセンサノードがどのモノに取り付けたのかを把握できない．そのため，モノとセンサノードIDを対応付ける必要がある．
2. **状況に応じた動作スイッチング**
   同一のサービスであっても，センサノードが置かれた状況によって稼動することが相応しくない場合がある．そのため，状況に応じてセンサノードの動作を切り替える必要がある．
3. **複数センサノードのグルーピング**
   落し物検知のように複数のセンサノードが連携して動作する場合，センサノードの数が増えると連携すべきセンサノードの管理が難しくなる．そのため，複数のセンサノードをグループとしてまとめる必要がある．
4. **外部資源との連携**
   センサノードが持つセンシング能力，アクチュエーション能力，計算能力は乏しい．そのため，サービスによってはセンサノードを外部資源と連携させる必要がある．

3.2 uElementで解決する課題

uElementは，センサノード用プラットフォームとして，前項で挙げた課題2及び課題3を解決する．外部資源上ではなくセンサノード上でこれらの課題を解決することによって，モノに取り付けられたセンサノード以外に資源がない場合でもサービスを提供可能になる．それぞれの課題を解決するため，uElementは動作スイッチング機能とグルーピング機能を持つ．

uElementの利用手順は以下の通りである．

(1) uElementが乗ったセンサノードをモノに取り付ける
(2) uElement上にサービスをインストールする
(3) 実際にuElementを稼動させ，サービスを利用する

uElementにサービスをインストールする際にPCやPDAが必要になるが，一度サービスをインストールすれば，次からはPCやPDAを用いなくてもサービスを利用できる．

4 機能要件

本節では，uElementの機能要件について述べる．uElementはセンサノード上で稼動し，状況に応じてセンサノードの動作を変更する動作スイッチング機能と，複数のセンサノードをまとめるグルーピング機能を持つ．まず動作スイッチングの機能要件について述べ，次にグルーピングの機能要件について述べる．

4.1 動作スイッチングの機能要件

動作スイッチングの機能要件を以下に示す．

- **シンプルな動作記述**
  PCやPDAからuElementに動作をインストールする際，センサネットワーク上を流れるデータ量を抑えるため，uElementによるサービスの動作定義はシンプルである必要がある．
- **状況に応じた動作スイッチ**
  状況に適したサービスを稼動させるため，動作切り替えを行う際に自ノードのセンサデータや周辺ノードの状況を考慮する必要がある．
- **計算資源のサービス間共有**
  センサノード上の計算資源が限られているため，uElementが複数のサービスを扱う場合，サービス間で可能な限り処理を共有する必要がある．

4.2 グルーピングの機能要件

グルーピングの機能要件を以下に示す．

- **自律的なグルーピング**
  中心ノードでグルーピングを行うという方法では，中心ノード自体がグループから外れてしまった場合に対処できない．そのため，グルーピングは各ノードで自律的に行う必要がある．
- **潜在的な情報によるグルーピング**
  uElementは外部資源を利用できない場所でも動作するため，グルーピングに用いる情報は，センサデータやノードIDのようにセンサノードが潜在的に持つ情報である必要がある．
- **時間幅を持ったグルーピング**
  即時的なグルーピングだけではなく，ある程度の期間を分析してグルーピングを行う必要がある．

5 設計

本節ではuElementの設計について述べる．uElementはセンサノードの動作変更を行う動作スイッチング機構 [9] と，複数のセンサノードでグルーピングを行うグルーピング機構 [16] で構成される．図2にuElementの全体構成図を示す．グルーピング機構と動作スイッチング機構と分けて設計することによって，動作スイッチング機構だけでなく，uElement上で動作するアプリケーションもグルーピング情報を利用可能になる．

各機構は連携して動作する．まずグルーピング機構はセンサ情報および他ノードのセンサ情報を用いてグルーピングを行い，その結果をグループ情報として動作スイッチング機構に渡す．次に動作スイッチング機構は受け取ったグループ情報から動作の切り替えを行い，必要に応じてアクチュエータに命令を送る．次項より各機構の設計について述べる．

5.1 動作スイッチング機構の設計

本節では，uElementの動作スイッチング機構の設計について述べる．

動作スイッチング機構の設計方針

本項では，動作スイッチング機構の設計について述べる．センサノード上で動作スイッチングを行うためには，まず各動作を定義し，センサ情報に応じてどの動作をアクティブにするかの決定を行う必要がある．一般的にセンサノードの動作は，センサによる入力と，アクチュエータによる出力，および無線による入出力によって構成される．その

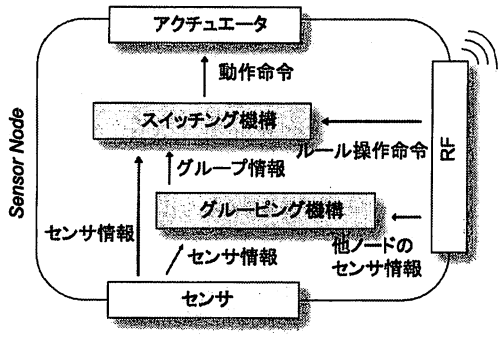

図 2: uElement の全体構成図

ため，センサノードの動作定義は，入力と出力を結びつけ，ある入力に応じて特定の出力が行われるようにすると直感的にわかり易く，実装も容易である．

そこで，動作スイッチング機構ではルールによる動作定義を行い，動作を Event と Action の組み合わせで構成されるルールで表現する．また，動作切り替えのポリシーもルールによって定義し，センサ情報やグルーピング情報の変化をアクティブなルールを切り替える Trigger として用いる．すなわち動作スイッチング機構は，動作定義ルールと動作切り替えルールを解析することによって動作切り替えを行う．さらにこの時，uElement に予め準備された Event や Action のみをルールに記述させることによって，センサノードの資源を節約する．尚，uElement ではこれらのルールを区別するため，Event と Action によって動作を定義するルールを Role，動作を切り替えるためのルールを Switching Rule(以後，SRule) と呼ぶ．

以上に述べた uElement における動作スイッチング機構の特徴を以下にまとめる．

(1) 動作定義方法

uElement では，動作を Event と Action によって定義することで，シンプルな動作定義を提供する．

(2) 動作切り替え方法

Trigger としてセンサ情報とグループ情報を利用することで，状況に応じた動作切り替えを行う．

(3) ルール評価方法

ルールを解析し，予め uElement 上に用意された機能を実行することで，効率的な資源利用を実現する．

また，図 3 に動作切り替えの概略図を示す．

ルールフォーマット

動作スイッチング機構で用いるルールのフォーマットを図 4 に示す．uElement では，Event 及び Action は予め uElement 上に用意されており，Role，Tuple，SRule は，それらの中から必要なルールを指定するという方法を用いた．尚，Trigger のフォーマットは Event と共通のため，図 4 では省略した．

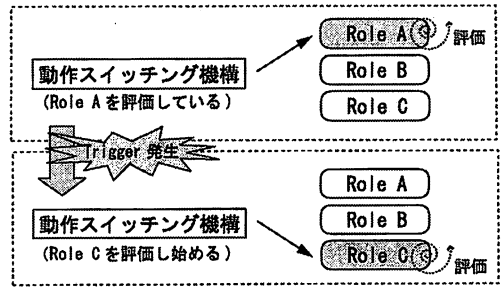

図 3: 動作切り替えの概略図

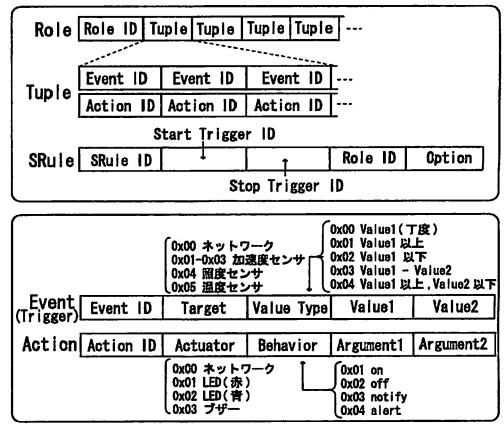

図 4: ルールフォーマット

動作概要

1. まずセンサ情報及びグルーピング情報を監視し，その値においてアクティブになるべき Role が存在するかどうかを SRule を用いて調べる
2. 1 において，もしアクティブになるべき Role がある場合は該当する Role のアクティブフラグを立てる．逆に，アクティブになるべきではない Role が存在した場合，該当する Role のアクティブフラグを解除する．
3. アクティブな Role が監視する Event を評価し，該当する Event が発生していると評価された場合は，Role 中の Tuple によって Event 対応付けられた Action を実行する．

設計図

動作スイッチング機構の設計図を図 5 に示す．動作スイッチング機構はセンサ情報処理部，スイッチング管理部，ルール管理部，及びルール DB によって構成される．以下に各部の概要を示す．

- **センサデータ処理部**

  センサ情報を取得し，外れ値の処理や平均の算出などの簡単なデータ処理を行った後，スイッチング管理部に処理したセンサデータを渡す．

- **スイッチング管理部**
  センサデータやグルーピング情報を受け取り，Role や SRule が格納されたルール DB から必要なルールを取り出して評価を行う．評価の結果によっては，アクチュエータに命令を送る．
- **ルール管理部**
  ルール DB に対しルールの追加や更新を行うことによって，外部からルールの操作を可能にする．

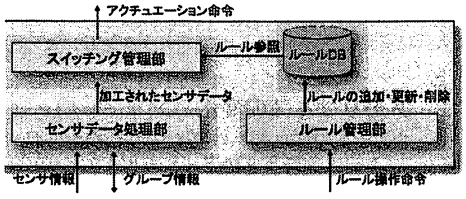

図 5: 動作スイッチング機構の構成図

5.2 グルーピングの設計

本節では，uElement のグルーピング機構について述べる．

グルーピング機構の設計方針

uElement のグルーピング機構は，与えられた条件に基づいてグループを作成し，グループに含まれるセンサノードの ID リストを返す．以下に，uElement におけるグルーピング機構の特徴をまとめる．

(1) グルーピング条件

uElement のグルーピング機構では，センサ情報をグルーピング条件として用いる．自ノードと周辺ノードにおけるセンサ情報差を求め，差が近いノード程，同一グループになる可能性が高いと判断する．

(2) グルーピングを行う場所

グルーピングは，各センサノードで行う．各々のセンサノードが周辺ノードとセンサ情報を共有し，自ノード上でグループを作成する．

(3) グルーピング情報の保存

グルーピングの過程で計算されたセンサ情報差を蓄積し，時間的な幅を持たせたグルーピングを行う．

グルーピング機構に対し，これらの特徴を持たせた理由について述べる．まず，外部資源を利用せず，センサノード上の情報だけでグルーピングを行うには，センサ情報に基づいたグルーピングが有効である．センサ情報だけでも，センサノードの置かれた物理的状況を把握できるため，有用なグループが作成できる．

また，各センサノード上でグルーピングを行うことにより，中心ノードが存在しなくてもグルーピングを利用したサービスが動作可能になる．そのため，中心ノードが欠落するような場面でも，サービスの継続提供が可能になる．

さらに，グルーピングの過程で計算されたセンサ情報差を蓄積することによって，時間的な幅を持たせたグルーピングが可能になる．このとき，センサ情報ではなく，計算されたセンサ情報差を保存することでメモリの節約を行う．

グルーピング方法

uElement では自ノード及び周辺ノードから得られたセンサ情報を用いてグループを作成する．センサ情報を指定してグループを作成する際，自ノードと物理的に同じような状況にあるセンサノードでグループを作成するためには，センサ情報の絶対値による指定だけではなく，相対値による指定も行える必要がある．そこで，uElement では特に相対値によるグルーピング作成に焦点を当て，自ノードのセンサ情報と周辺ノードのセンサ情報との差をグルーピング条件として用いた．

センサ情報の差はユークリッド距離として求められる．a と b はセンサノード a 及び b を表し，この二つのセンサデータの距離 sd は，以下の式によって求められる．

$$sd(a,b) = \sqrt{(x_a - x_b)^2}$$

この計算を周辺ノードすべてに対して行い，自ノードと各ノードとの距離を算出する．

動作概要

グルーピング機構の動作概要を以下に示す．

1. 自ノードのセンサデータを取得し，さらにそれを周辺ノードと共有する．
2. 各周辺ノードのセンサデータと自ノードのセンサデータとの差を求めて保存する．このとき，複数の周辺センサノードと一元的に比較するため，各周辺センサノードとの差をセンサデータのユークリッド距離として求める．
3. 2 の結果得られた複数のユークリッド距離を用いてクラスター分析を行い，グループのリストを作成する．

設計図

グルーピング機構の設計図を図 6 に示す．グルーピング機構はセンサ情報処理部，履歴情報生成部，グループ抽出部，通信管理部によって構成される．以下に各部の概要を示す．

- **センサ情報処理部**
  センサ情報を取得し，外れ値の除外や平均の算出などを行う．
- **履歴情報生成部**
  各センサ情報から距離を算出し，それらを履歴として保存する．
- **グループ抽出部**
  外部から抽出条件が与えられると履歴から条件に該当するセンサの距離情報を取得してグループを作成し，その結果をグループ情報として返す．
- **通信管理部**
  各部の処理における他ノードとの通信を統括する．

6 プロトタイプ

本節では，本論文で実装したプロトタイプについて述べる．

6.1 実装環境

プロトタイプの実装環境として，センサノードとして Smart-Its Particle 2/29(図 7) を用いた．Smart-Its Particle

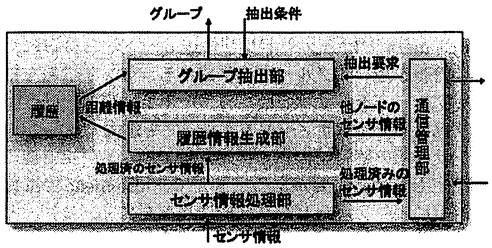

図 6: グルーピング機構の構成図

2/29 の仕様を表 1 に示す．プログラムは C 言語で作成し，コンパイラには CCS 社の PIC C コンパイラ [2] を用いた．

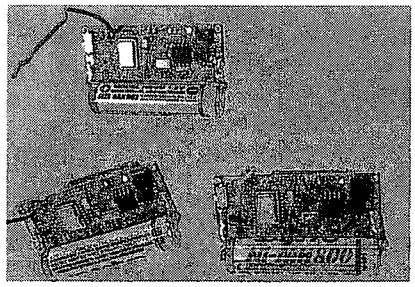

図 7: 実装環境 (Smart-Its Particle 2/29)

CPU	PIC18F6720
Program Memory	128Kbyte
RAM	4Kbyte
Data EEPROM	1Kbyte

表 1: Smart-Its Particle 2/29 の仕様

6.2 実装方針

uElement の実装量を最小限にするため，Particle Base System を利用した．Particle Base System は，Particle 上のアプリケーションが不必要なパケットを拾ってしまうことを防ぐためのネットワークプロトコルである AwareCon[3] を提供する．uElement もネットワークプロトコルの実装を省くため，AwareCon を利用した．図 8 に uElement と AwareCon の関係を示す．

また，動作定義ルールやセンサデータの蓄積に伴うメモリ管理のコストを減らすため，動作定義ルールやセンサデータに用いるメモリ空間を固定長配列として実装した．

7 評価

本節ではプロトタイプの評価について述べる．評価項目として，まず uElement を実装することによって生じたメモリサイズの変化を挙げる．また，他ノードとの通信が増えることによって生じたパケット量の変化，及び uElement

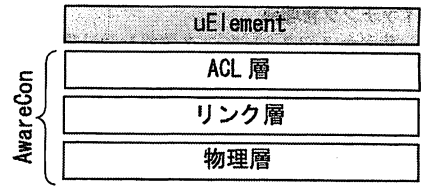

図 8: AwareCon における uElement の実装

の実行に必要となる時間とインストラクション数の変化も評価項目として挙げる．

7.1 メモリサイズの変化

センサノードのプログラムメモリサイズは小さいため，uElement を組み込むことによって生じるメモリサイズの増加量は重要である．そこで，Particle の Base System に uElement を加える前と後でメモリサイズを比較した．メモリサイズは，CCS C コンパイラによって生成される焼き込み用ファイルのデータ部分のみを合計して求めた．

比較結果を図 9 に示す．まず，Particle の Base System のみを含んだメモリサイズは 45,552 バイトであった．また，この Base System にグルーピング機構のみを加えたメモリサイズは，54,176 バイトであり，増加サイズは 8,624 バイト，Base System に対する増加率は約 19%であった．さらに動作スイッチング機構を加えた uElement 全体のメモリサイズは 57,904 バイトであり，Base System に対する増加サイズは 12,352 バイト，増加率は約 27%であった．以上の結果から，uElement が Particle Base System に対して占める割合は小さく，メモリサイズの小さいセンサノード上でも十分に実現可能であると言える．

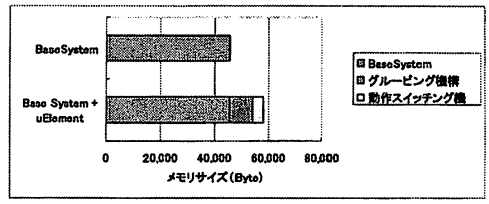

図 9: メモリサイズの増加量

7.2 送受信するパケット量の変化

グルーピング機構はグルーピングを行う際に，他ノードと通信を行う．この通信において，ノード数が増えた場合のスケーラビリティを把握するため，ノード数が増えることで送受信するパケット量にどのような変化があるのかを計測した．計測のために受信のみを行う sink ノードを用意し，センサネットワークに流れるパケット数を計測した．計測結果はセンサネットワークに流れたパケット数を一分間計測し，毎秒ごとのパケット数を求めた．また，実際に得られた計測結果が理論的な値とどの程度離れているのかを把握するため，単一ノードで得られた結果をノード数倍

した値を理論値として求めた．尚，ノード数は1から4までで計測した．

計測結果および理論値を図10に示す．1ノード増えた場合のパケット数は毎秒2.9パケットであった．この値をもとに理論値を計算したところ，実測値は理論値と同様にほぼ線形で増加した．但し，ノード数4においてパケット数の減少が起きたが，これは，センサネットワークを流れるパケット数が減少したのではなく，無線干渉が起き，結果的にsinkノードで計測できたパケット数が減少したためと考えられる．以上の結果から，ノード数の増加によるパケット数の増加率はスケールするが，ノード数が多くなる場合は干渉を防ぐ必要があると言える．

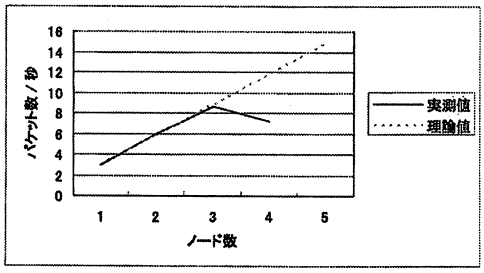

図 10: パケットの増加量

7.3 実行時間の変化

uElementによって実行速度がどの程度増加したのかを計測した．Particle Base Systemは同じ処理を繰り返し行うことによってセンサノードを制御するため，実行時間としてこの繰り返しを一周する時間を計測した．計測結果を図11に示す．Particle Base System単体での実行時間は340ミリ秒であった．これにグルーピング機構を加えた場合の実行時間は1212ミリ秒，さらに動作スイッチング機構を加えた場合は1924ミリ秒であった．この実行時間増加の原因は，パケット送信後に必ず必要となる待ち時間や動作ルールの評価であると考えられる．

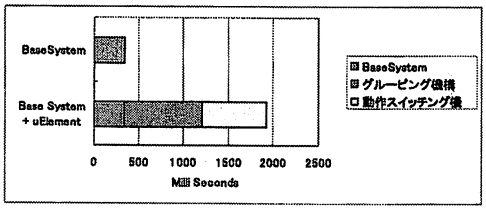

図 11: 実行時間の増加量

8 今後の課題

本節では今後の課題について述べる．今後の課題は大きく分けて，第3節で述べた課題の中で本論文では扱わなかった他の課題へのアプローチと，今回プロトタイプとして実装したuElementの課題解決の二点である．まず他の課題へのアプローチについて述べ，次にuElementの機能拡張について述べる．

他の課題へのアプローチ

我々のuTopiaプロジェクトでは「モノとセンサノードの対応付け」を解決するため，uAssociator[14]の研究を行っている．uAssociatorは，モノにセンサノードを取り付け，モノの写真を撮ることによって写真とセンサノードIDの対応付けを行うアソシエーション機構である．

また，「外部資源の連携」についても研究を開始しており，uElementの拡張およびセンサネットワークと環境側ネットワークとのブリッジ機構を開発中である．

uElementの課題解決

uElementの課題として，以下の点が挙げられる．

- ルール書き込み用ユーザインタフェースの作成
- グルーピングの性能向上
- 複数センサノードにおける動作の整合性保障

一点目としてルール書き込み用ユーザインタフェースの作成を行う必要がある．プロトタイプとして実装したuElementでは，動作スイッチング機構で用いるRoleやSwitching Ruleをセンサノード上に直接書き込んでいたため，このままではユーザに対してルールを作成する知識や能力を要求することになり現実的ではない．そこでuAssociatorとの連携を考慮し，GUIの書き込みツールを作成する．

二点目として，センサノードをグルーピングする際の性能を向上させる必要がある．今回実装したグルーピング機構は，予め設定された間隔でセンサデータを送信していたため，ノード数が増えることによって無線電波の干渉が発生した．この問題を解決するため，周辺ノードの数に応じて送信間隔を変更する必要がある．また，過去のセンサデータを圧縮するなどして，保存すべきデータのサイズを抑える必要がある．

三点目として，複数センサノードで動作スイッチングを行う際の動作整合性を保障する必要がある．もし複数のセンサノードで連携することを前提に書かれたRoleが与えられた場合，動作スイッチングを行うことによって整合性が取れなくなる可能性がある．この問題を解決するためには，動作スイッチングを行うときに周辺ノードと通信を行い，あるポリシーに基づいて周辺ノードと共に動作を切り替える必要がある．

9 結論

本論文では，ユーザ自身によるモノの知的化に着目し，低コストな知的化を可能にするためセンサノードを用いた．センサノードによるモノの知的化には汎用的なプラットフォームが必要であり，本論文ではまずセンサノードを利用したモノの知的化を実現するための課題を4つ挙げ，次に複数のセンサノードによる状況に応じたサービス提供を可能にするセンサノード用プラットフォーム「uElement」について述べた．

uElementは，動作スイッチング機構とグルーピング機構によって構成される．動作スイッチング機構は，センサノードの動作定義をルールによって行い，そのルールをセンサ情報とグループ情報によって切り替えることで動作スイッチングを実現する．一方，グルーピング機構はセンサデータを周辺ノードと共有し，自ノードと周辺ノードのセンサデータ差を計算し，クラスター分析を行うことでセンサノードのグルーピングを実現する．

本論文ではSmart-Its Particleを利用してmiddlewareのプロトタイプを作成した．評価の結果，uElementを実装することによるメモリサイズの増加量は12,352バイトであり，周辺ノードが1ノード増えるごとの通信パケット数増加量は2.9パケット/秒であった．また，1サイクルの実行時間増加量は約1.6秒であった．これらの結果により，uElementがセンサノード上で十分動作可能であることを示した．

今後の課題として，middlewareの改善と共に，モノとセンサノードの対応付けを行うuAssociatorとの連携や，外部資源を利用したサービス提供を行う．

謝辞

この研究は総務省「ユビキタスネットワーク制御・管理技術の研究開発（ubilaプロジェクト）」の一部として行われました。

参考文献

[1] Crossbow technology inc. http://www.xbow.com/.

[2] Custome computer services, inc. http://www.ccsinfo.com/.

[3] M. Beigl, A. Krohn, T. Zimmer, C. Decker, and P. Robinson. Awarecon: Situation aware context communication. In *the Fifth International Conference on Ubiquitous Computing (UbiComp)*, Seattle, USA, Oct. 2003.

[4] Michael Beigl, Christian Decker, Albert Krohn, Till Riedel, and Tobias Zimmer. μ parts: Low cost sensor networks at scale. In *Ubicomp Demo Session*, 2005.

[5] Michael Beigl, Hans-W. Gellersen, and Albrecht Schmidt. Mediacups: experience with design and use of computer-augmented everyday artifacts. *Computer Networks (Amsterdam, Netherlands: 1999)*, Vol. 35, No. 4, pp. 401–409, 2001.

[6] Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, and Steven A. Shafer. Easyliving: Technologies for intelligent environments. In *HUC*, pp. 12–29, 2000.

[7] Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *International Conference on Ubiquitous Computing(UbiComp)*, 2001.

[8] Masaki Ito, Akiko Iwaya, Masato Saito, Kenichi Nakanishi, Kenta Matsumiya, Jin Nakazawa, Nobuhiko Nishio, Kazunori Takashio, and Hideyuki Tokuda. Smart furniture: Improvising ubiquitous hot-spot environment. pp. 248–253, May 2003.

[9] Kengo Koizumi, Hiroshi Sakakibara, Masayuki Iwai, and Hideyuki Tokuda. A context-oriented application switching mechanism for daily life supports. In *The Seventh International Conference on Ubiquitous Computing (Ubicomp2005) Poster Session*, Tokyo, Japan, Sep. 2005.

[10] Mik Lamming and Denis Bohm. Specs: Another approach to human context and activity sensing research, using tiny peer-to-peer wireless computers. In *International Conference on Ubiquitous Computing(UbiComp)*, 2003.

[11] T. Okoshi, S. Wakayama, Y. Sugita, S. Aoki, T. Iwamoto, J. Nakazawa, D. Furusaka, M. Iwai, and A. Kusumoto. Smart space laboratory project: Toward the next generation computing environment. *IEEE International Workshop on Networked Appliances (IWNA'01)*, pp. 115–121, March 2001.

[12] Martin Strohbach, Hans-Werner Gellersen, Gerd Kortuem, and Christian Kray. Cooperative artefacts: Assessing real world situations with embedded technology. In *International Conference on Ubiquitous Computing(Ubicomp)*, 2004.

[13] Mark Weiser. The computer for the 21st century. pp. 933–940, 1995.

[14] Takuro Yonezawa, Hiroshi Sakakibara, Kazunori Takashio, and Hideyuki Tokuda. uassociator: A spotlight-and-camera mapping tool for associating sensor nodes and everyday objects. In *Proceedings of The 4th International Conference on Pervasive Computing (PERVASIVE2006) Demo*, May 2006.

[15] 椎尾一郎, Jim Rowan, Elizabeth Mynatt. Digital decor: 日用品コンピューティング. ヒューマンインタフェース学会論文誌, Vol. 5, No. 3, pp. 323(11) – 330(18), 2003.

[16] 中井彦一郎, 出内将夫, 榊原寛, 船木康平, 徳田英幸. サービス支援型センサネットワークグルーピング機構. 情報処理学会システムソフトウェアとオペレーティングシステム研究会ポスターセッション, 5月 2005.