

# センサリッチ携帯端末における コンテキスト取得を支援するミドルウェア開発

花岡 健介 高木 綾子 田村 真浩

{kensuk,ayako,tamura}@dcl.info.waseda.ac.jp

早稲田大学大学院 理工学研究科 情報ネットワーク専攻

**概要:** ユビキタスコンピューティング環境において、実空間を認識するコンテキストウェア技術は重要である。しかしながら、多種のセンサを搭載した個人端末によるコンテキスト取得に関しては議論が十分になされていない。本研究では15種類のセンサを搭載した個人端末 Muffin を用いてコンテキスト取得における問題点を指摘し、コンテキスト取得を支援するミドルウェアへの要求事項を考察する。そして考察に基づき、コンテキストウェアなアプリケーションに対してコンテキスト取得を支援するミドルウェア Cinnamon を開発する。

**キーワード:** コンテキストウェアネス, ミドルウェア, センサリッチ携帯端末

## Cinnamon: Context Information Acquisition Middleware for Sensor-rich Devices

Kensuke Hanaoka Ayako Takagi Masahiro Tamura

{kensuk,ayako,tamura}@dcl.info.waseda.ac.jp

Graduate School of Science and Engineering, Waseda University

**abstract:** In a ubiquitous computing environment, context awareness is significant. However, acquiring the context from the sensor equipped devices has not been discussed sufficiently. Therefore, we identify the problems of acquiring the context from Muffin, which is the personal device with 15 kinds of sensors, and consider the requirements of the context aware middleware. As a result, we develop the middleware named "Cinnamon", which satisfies the requirements and provide the context to the context aware application.

**Keywords:** Context Awareness, Middleware, Sensor-rich Device

### 1 序論

無線通信技術の発達や計算機の小型・高性能化によって計算機やセンサは日常空間に遍在するようになり、日常生活で我々を取り巻くサービスは実空間の情報(コンテキスト)を取得することが可能になる。このような環境はユビキタスコンピューティング環境と呼ばれ、コンテキストを利用してユーザの状況(ユーザコンテキスト)に適応したサービスを提供するコンテキストウェア技術は重要となる [1]。

また、PDA や携帯電話のようなパーソナルデバイス

は、様々なコンテキストを知覚できるように拡張されつつある [2][5]。パーソナルデバイスはユーザと密接に関係しているため、状況に応じ適切に振舞うことを期待される。しかし、パーソナルデバイスを用いたコンテキスト解析とセンサの適性に関する研究例や、解析したコンテキストを用いたアプリケーション例、解析結果の分解能まで踏み込んだ研究例が少ないなど、パーソナルデバイスを用いたコンテキスト取得に関する考察が少ないという問題点が挙げられる。

そこで、本研究では15種類のセンサを搭載した次世代携帯端末 Muffin を用いて、観測可能なコンテキスト

とそれぞれのセンサの適性について考察する。その考察に基づき、個人端末におけるコンテキスト取得への要求事項を示し、コンテキストアウェアなアプリケーションにコンテキストを提供するミドルウェア Cinnamon の開発を行うことが研究の目的である。

以下では、まず2章で関連研究を述べる。3章で次世代センサリッチ携帯端末 Muffin を紹介し、4章で Muffin におけるコンテキスト取得への要求事項を説明する。そして5章で要求事項に基づいた Cinnamon の設計、6章で Cinnamon 上で動作するアプリケーション例、最後に7章で結論および将来課題を述べる。

## 2 関連研究

携帯端末にセンサを組み込んだ研究として、Ken Hinckley らの PDA のインタフェースの拡張がある [6]。この研究では、PDA 内に2軸加速度センサ、赤外線センサ、タッチセンサを組み込み、アプリケーションのユーザビリティを向上させている。例えば、ディスプレイの向きに合わせて画面を切り替えるために、2軸加速度センサから“ディスプレイの向き”を取得する。また、赤外線センサによって得られる“デバイスに顔が近づいている”というコンテキストをきっかけに、ボイスメモアプリケーションを実行する例も示している。その他、タッチセンサから“デバイスを握っている”というコンテキストを、2軸加速度センサから、“デバイスを見ている”、“ユーザが歩いている”、“ディスプレイの向き”というコンテキストを取得している。

Marmasse らは腕時計型のコンテキストアウェアなコミュニケーションである WatchMe を開発している [7]。WatchMe では、GPS、3軸加速度センサ、マイクロフォンを搭載したデバイスをユーザが携帯し、位置・移動・会話の状態を取得する。取得されたコンテキストがアイコンを用いて表現できる場合、それらのアイコンがユーザに関する写真画像と共に WatchMe 上へ表される。また、これらの情報は他者と共有することができる。自身情報を取得している他者についても知ることができる。WatchMe ではメール、音声メッセージ、通話の3種類のコミュニケーション方法をサポートしており、コミュニケーションをとりたい相手のコンテキストを知ることで、最適なコミュニケーション手段を選択できる。

コンテキストアウェアなミドルウェアの研究としては、Daniel Salber らによる Context Toolkit がある [3]。Context Toolkit はユーザに対してコンテキストアウェア

なアプリケーションを提供するためのソフトウェアツールキットである。設計は我々のミドルウェアと似ている部分もあるが、コンテキストアウェアなアプリケーションを開発する際に GUI ツールを使わなければならない、柔軟性に欠けると言える。

Korpiainen らは、我々と同様に多種のセンサを搭載した携帯端末からのコンテキスト取得を支援するフレームワークを提案している [4]。センサだけでなく、ネットワークやデバイスの状態からもコンテキストを抽出しているのが特徴である。そして、抽出したコンテキストがどの程度信頼できるかという精度をコンテキスト情報に含めているが、精度をフレームワーク内でどのように使い、アプリケーションにどのような利益をもたらすかについては言及していない。

## 3 Muffin とコンテキスト取得

### 3.1 次世代センサ端末 Muffin

Muffin は、早稲田大学中島研究室とノキア・ジャパンとの共同研究で開発された、ユビキタスコンピューティング環境における研究・実験のための次世代センサデバイスである。Muffin と搭載されている15種類のセンサの関係を図1に示す。

Muffin に搭載されているセンサは4つのカテゴリに分類することができる。1つ目の環境センサには、温度センサ、湿度センサ、気圧センサが含まれる。2つ目の生体センサには、心拍センサ、皮膚抵抗センサ、皮膚温度センサ、アルコールセンサが含まれる。3つ目のモーション・位置センサには、3軸加速度センサ、コンパス、グリップセンサ、GPS、超音波距離測定器が含まれる。そして最後に RFID 検出器、カメラ、マイクが搭載されている。図2のように Muffin を手に握ることで全てのセンサによる計測ができることが特徴である。

Muffin はセンサだけでなく、無線 LAN、Bluetooth、RS232 など各種無線・有線のインタフェースも装備している。これらのインタフェースを通して、他のコンピュータと通信することができる。

Muffin には Linux が搭載されており、デバイスファイル (e.g. /dev/sensor/Pulse) に対してシステムコール open(), read(), close() を実行することでセンサデータを取得することができる。

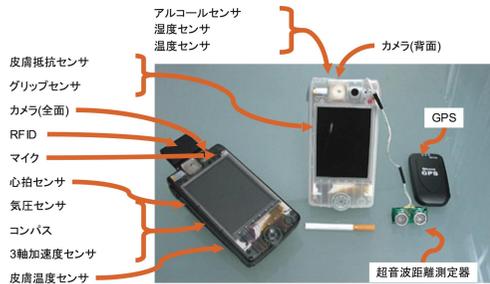


図 1: Muffin と 15 種類のセンサ



図 2: Muffin の利用例

### 3.2 Muffin におけるコンテキスト取得

Muffin は 15 種類ものセンサを搭載しているため、ユーザの状況、環境情報など多くのコンテキストを取得できると期待されるが、Muffin のようなセンサリッチ端末において実際にどのようなコンテキストを取得できるかはいまだ明確にされていない。そのため、Muffin から実際にデータを取得し、観測可能と考えられるコンテキストと解析に有効なコンテキストを考察した。その結果、ユーザコンテキスト取得にはモーション・位置センサを用いることが有効であった。例えば、3 軸加速度センサのデータを解析することで“Muffin の向き”や“ユーザが歩いているか否か”というコンテキストを得ることができる。そして、皮膚抵抗センサやグリップセンサを用いることで、“ユーザが Muffin を手にしているか否か”を判別することができる。

## 4 コンテキスト取得への要求事項

本章では Muffin におけるコンテキスト取得への要求事項を考察する。4.1 節では高抽象度のコンテキストを取得するための機構としてコンテキストを段階的に抽象化する手法を紹介する。そして、4.2 節で 1 つのコンテキスト取得に対して複数の解析アルゴリズムを用い、

状況に応じて最適な解析アルゴリズムを選択する必要性を述べる。

### 4.1 段階的抽象化

高抽象度のコンテキストを生成するには、抽象度の低いコンテキストから段階的に抽象度を上げていく必要がある [8]。なぜなら、高抽象度のコンテキストを 1 つの解析アルゴリズムにより生成すると、解析アルゴリズムのコードが肥大化、複雑化してしまうからである。そして、“ユーザが立っている、座っている”という状況の認識でさえ、実質的な難しさがあるとわかった。その理由は、ユーザによって Muffin の持ち方や持った際の姿勢が異なるため、コンテキスト解析の複雑さが増すからだと考えられる。この問題を回避するためにも、コンテキストは細かい単位で扱われるべきである。もし抽象度の高いコンテキストを“Muffin が動いている”という単純なコンテキストに分解できるなら、単純なコンテキストから複雑なコンテキストを構築するのは簡単である。例えば、“ユーザが Muffin を持って歩いている”というコンテキストの生成は、“Muffin を握っている”, “Muffin が動いている”という単純な 2 つのコンテキストから生成されるべきである。図 3 に段階的に抽象度を高めていく例を示す。

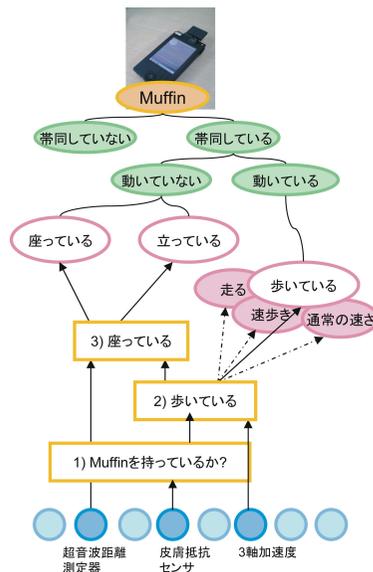


図 3: 段階的抽象化の例

## 4.2 最適な解析アルゴリズムの選択

あるコンテキストを取得するための解析手法は1つとは限らない。さまざまな観点から解析し、一番有効な解析を選択する必要がある。そこで我々は“ユーザーが歩いているか?”を解析するために閾値とFFT(Fast Fourier Transform)の2種類の解析を行った。閾値の解析アルゴリズムでは、“歩いている”、“止まっている”のみの判定をしている。FFTを用いた解析方法は、“歩いている”、“止まっている”の判定のみに留まらず、歩く速さを3段階で表すことが可能である。しかし、FFTの解析アルゴリズムは多くのデータを必要とするため、“歩いている”と“止まっている”の状態遷移を反映するのに平均5.55秒かかる。

状態遷移が起きた際にタイムスタンプを取得し、状態遷移の判定に要する時間、および状態認識率を求めた。状態認識率は、 $\text{状態認識率}(\%) = \frac{\text{正しく認識した時間}}{\text{実際の動作時間}} \times 100$  で求める。表1に判定時間、表2に状態認識率を示す。

表 1: 判定時間 (単位:秒)

解析方法	walking → resting	resting → walking
閾値ベース	1.0	1.0
FFT ベース	5.7	5.4

表 2: 状態認識率

解析方法	walking	sitting	standing
閾値ベース	82 %	79 %	92 %
FFT ベース	93 %	85 %	99 %

2つの解析方法には精度・分解能と判定時間においてそれぞれ長所・短所がある。特に、精度・分解能と判定時間の間にトレードオフが存在する。ユーザーコンテキストを利用するアプリケーションにより、ユーザーコンテキスト解析に求める精度・性能は異なる。アプリケーションに応じて用いる解析手法を選択することが重要だと考えられ、そのような機能を提供する機構が必要である。

状況により有効なリソースが変化する場合がある。解析に用いるリソースが無効になった際、代替するリソースを用いた別の解析アルゴリズムが必要である。状況に応じ、最適な解析アルゴリズムを提供できる機構が必要である。

## 5 設計

コンテキスト取得におけるこれらの要求事項を満たすミドルウェアCinnamonを開発した。ミドルウェア

のアーキテクチャを図4に示す。

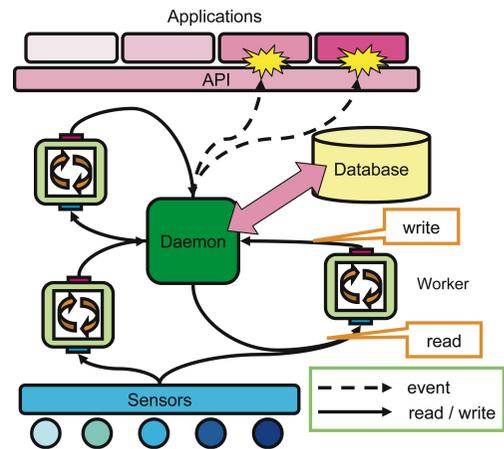


図 4: アーキテクチャ

CinnamonはWorker, Database, Daemonという3つの要素から構成する。以下にそれぞれの機能を述べる。

### Worker

Workerはセンサデータを解析し、コンテキストを生成する。例えばユーザーがMuffinを手をしているかどうかは、グリップセンサを閾値ベースで解析し判別する。Workerはセンサデータだけでなく、他のWorkerが生成したコンテキストを解析することで、抽象度のより高いコンテキストを生成する。例えばユーザーがMuffinの画面を見ているかどうかは、Muffinの向きというコンテキストとユーザーがMuffinを手をしているかどうかというコンテキストによって判別する。Muffinが上向きでありかつユーザーがMuffinを手にしていれば、ユーザーは画面を見ているというコンテキストを生成する。

### Database

Workerの生成したコンテキストを管理する。データベースには、組み込み向けデータベースSQLiteを用いた[9]。データベースで管理するコンテキストのフォーマットを表5に示す。

表 5: コンテキスト情報のフォーマット

```
Context := {Subject, Predicate, Worker ID,
            Time, Lag, Device ID, Confidence}
```

コンテキスト情報は、主語 (Subject), 述語 (Predicate) だけでなく、コンテキストを生成した Worker

表 3: データベース内のコンテキストの例

Subject	Predicate	Worker ID	Time	Lag	Confidence
User:Activity:Walking	resting	accel_walk	555	0	0.8
User:Activity:Direction	NW	direction	556	0	0.7
Device:Activity:TopSide	front	topside	556	0	0.9
User:Activity:Walking	walking	gps_walk	557	0	1.0
User:Activity:WalkingSpeed	fast	walking_speed	557	10	0.7
Environment:Temperature	hot	temperature	558	0	0.9

表 4: Cinnamon の提供する API

```
context_t* get_context(const char* subject);
int add_event_handler(const char* subject, void (*handler)(char*));
void remove_event_handler(void);
```

の識別子 (Worker ID), コンテキストを取得した時間 (Time), 解析に要した遅延時間 (Lag), コンテキストの信頼度 (Confidence) の 6 つの要素によって構成される。表 3 に Database 内で管理しているコンテキストの例を示す。

### Daemon

Worker, Database, そしてアプリケーションとのデータ通信を担う。そして, Worker の生成したコンテキストが変化した場合に, アプリケーションにイベント通知する機能を提供する。

コンテキスト情報の Confidence をもとに, 信頼度の高いコンテキストをアプリケーションに提供するのが Daemon の最大の特徴である。要求のあったコンテキストを Database から探し, ある一定の時間内でもっとも Confidence の高いコンテキストを選択する。Confidence はセンサの有効性および判定アルゴリズムの正確さをもとに算出される。センサの有効性はセンサが有効な値を示しているかどうかによって決定し, 判定アルゴリズムの正確さは Worker の開発者が決定する。この Daemon の機能により, 一つのコンテキスト解析に対して複数の解析アルゴリズムがあっても常に最適なコンテキストを得ることができる。

Cinnamon にはブラックボードアーキテクチャを採用した。各 Worker は他の Worker に依存せず独立して動作し, 非同期的に Database へ読み書きを行うことができる。新たなコンテキストをアプリケーションに提供する場合に, 容易に Worker を追加することができるのが特徴である。

アプリケーションには Database 内のコンテキストを

取得するための API を提供する。アプリケーション開発の幅を広げるために, コンテキストを取得するにはイベントおよびクエリによる 2 種類のインタフェースを用意する。アプリケーションに提供する API を表 4 にまとめた。

## 6 アプリケーション例

本章では, Cinnamon を利用して開発するアプリケーションについて述べる。

### IM(インスタントメッセージング) アプリケーション

ネットワーク上で自分の仲間の状態を知り, メッセージ交換などをおこなうインスタントメッセージング (IM) サービスを考える。相手の状態を知ることができるのが有用な点である一方で, ユーザが手動で自分の状態を更新しなくてはならないという煩わしさがある。モバイル環境を考えると, この操作はさらにユーザにとって負担の大きなものになる。そこで, ミドルウェアが取得可能なユーザコンテキストをマッピングすることで, 自動的に状態を変更できるようになる。そのほかにも, ユーザが外を歩いていたたり, 周りがうるさかったりというコンテキストを取得した際には, メッセージ通知に音声を利用する。一方で, ユーザのスケジュールから会議中や授業中といったコンテキストを取得できれば, 音声を使わずにバイブレーションで通知したり, 会議が終了してからメッセージを表示したりする。ユーザが端末を手にするというコンテキストを取得できるまで待つて, メッセージ通知を行うといったこともできる。

## リマインダ

次に、ユーザが設定した曜日や特定の時間に、自分に対してメッセージを通知するリマインダー機能を拡張させることを考えてみる。時間だけでなく、場所や会う人などを起動イベントとすることで、より有用な通知を行うことができる。例えば、セールをやっている店を知ったときに、その店の近くを訪れた際にその情報を通知するようにしておく。このような場合、何時訪れるかは分からなくても、その場所の近くにいるというコンテキストが有効である。そして、メッセージ通知の方法も IM サービスで利用したように、コンテキストに応じて使い分けることができる。

## 日記サポートアプリケーション

近年、Blog や SNS(Social Networking Site) などに自分の日記を公開する人が増えている。しかし、ある一日の全ての出来事を把握できる人は少なく、日記を書くときにその日何をしたか思い出すのに苦労する人も多いと思われる。そのようなときに、日記を書く手助けをするアプリケーションとして日記サポートアプリケーションを提案する。Cinnamon ではコンテキストを Database により管理しているため、何日分ものコンテキストをためることができる。日記を書くときにその日のコンテキストをユーザの見やすい形で提供する。例えば、その日歩いた軌跡を地図上に表示し、軌跡上の一点をクリックするとその場で取得されたコンテキストを表示する。さらに Muffin にはカメラが搭載されているので、コンテキストだけでなく撮影した写真とその時のコンテキストを組み合わせることで、さらにユーザを助けることができると考える。

## 7 結論・将来課題

本研究では、センサリッチな個人端末におけるコンテキスト取得の要求事項を考察し、15種類のセンサを搭載した個人端末 Muffin を用いてコンテキスト取得を支援するミドルウェア Cinnamon を開発した。そして、Cinnamon 上で動作するコンテキストアウェアなアプリケーション例を示した。今後はまずミドルウェアの性能を評価し、ミドルウェアの有用性を示していきたい。また、より抽象度の高いコンテキストをサポートするための機構が必要となってくる。そのためには Muffin だけでなく、他のリソースを活用しなければならないと考える。

## 参考文献

- [1] D.Siewiorek, A.Smailagic, J.Furukawa, N.Moraveji, K.Reiger, J.Shaffer, "SenSay: A Context-Aware Mobile Phone", In Proceedings of ISWC '03, 2003, pp. 248-249.
- [2] Chen G and Kotz D, "A Survey of Context-Aware Mobile Computing Research", Technical Report TR2000-381, Department of Computer Science, Dartmouth College,2000
- [3] Daniel Salber, Anind K.Dey and Gregory D.Adowb, "The Context Toolkit: Aiding the Development of Context-Enabled Applications", Conference on Human Factors in Computing Systems (CHI'99) pp.434-441
- [4] Korpipaa P, Mantyjarvi J, Kela J, Keranen H and Malm E-J, "Managing context information in mobile devices", IEEE Pervasive Computing 2(3):42-51.
- [5] Dirk Husemann, Chandra Narayanaswami, Michael Nidd, "Personal Mobile Hub", ISWC 2004
- [6] Hinckley K., Pierce, J., Sinclair, M. and Horvitz, E. "Sensing techniques for mobile interaction", Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST 2000),pp. 91-100.
- [7] N.Marmasse, C.Schmandt, D.Spectre, "WatchMe:Communication and Awareness Between Members of a Closely-Knit Group.", Ubicomp 2004, pp.241-231.
- [8] 藤波香織, 中島達夫, "コンテキストアウェアなアプリケーション構築のためのフレームワーク", 情報処理学会論文誌: コンピューティングシステム (ACS), pp.107-118, Vol.44, No.SIG10(ACS2), July, 2003.
- [9] SQLite: <http://www.sqlite.org/>