

## センサデータからコンテキスト抽出を行うための ビデオ動画を利用したコンテキスト手動付与ツール

佐々木 弘<sup>†</sup> 坂根 裕<sup>††</sup>  
太田 剛<sup>††</sup> 水野 忠則<sup>††</sup>

本稿では、筆者らがこれまでに構築したセンサアプリケーション開発環境「SATSUKI」を拡張し、センサを設置した環境の状況を開発アプリケーションに理解させるための枠組み「SATSUKI Debugger」を提案する。システム実現の第一歩として、アプリケーションで得られるセンサ系列に対し、環境に設置したカメラ映像からコンテキストを手動で割り当てるツールを実装し、学内にて実験を行った。実験データから、提案手法のいくつかの問題点と有効性に関する知見が得られた。

### A Tool for Manually Extracting Contexts Corresponding to Sensor Data from Video Images

HIROSHI SASAKI,<sup>†</sup> YUTAKA SAKANE,<sup>††</sup> TSUYOSHI OHTA<sup>††</sup>  
and TADANORI MIZUNO<sup>††</sup>

We have developed a sensor application development platform 'SATSUKI'. In this paper, we propose 'SATSUKI Debugger' for sensor application development understanding contexts of the environment. As a first step, we have developed a tool which enables developers to define contexts to sensor data through a video installed in the environment, and experimented in it. The experimental results suggest pieces of knowledge about both problem and validity of the proposed tool.

#### 1. はじめに

最近のユビキタスに関する研究の主要テーマは、機器操作のためのインタフェース研究から、環境を含めた人間活動のサポートに関する研究に移行しつつある<sup>1)~5)</sup>。このような研究では、対象となる人間活動をモデル化することや、環境や人体に装着したセンサからデータを収集しモデルとして蓄積すること、蓄積したデータを解析しサービスとして実世界にフィードバックすることが中心となる。

筆者らはこれまでに、環境や人体に設置した種々のセンサやアクチュエータを組み合わせ、手軽に必要なアプリケーションが作成できる環境として、SATSUKI<sup>6)</sup>を提案した。SATSUKIを利用することで、アプリケーション開発者は、センサドライバやネットワークを介したセンサ制御、複数センサの連携といっ

た敷居の高い技術に時間を浪費することなく、アプリケーションロジックの設計に集中できるようになった。しかし、細やかなサポートを行うサービスを実現するには、実世界で行われているコンテキストを理解できるモデル構築が鍵となるが、SATSUKIにはこの作業を支援する機能は存在していない。

本研究では、現実世界のモデルを構築せず「ほどほど」に細やかなコンテキスト抽出機能を実現することで、アプリケーション開発者の「現実世界モデルの構築コストを排除」し、「実装したアプリケーションの機能拡充」を狙う。環境にセンサ群を設置し、それらのセンサを利用したアプリケーションを実装した状況で、カメラやマイクから得られる実世界でのイベントと、センサデータ系列に対し、手動でコンテキストを関連付け知識として蓄積する。SATSUKIは、得られた知識を利用し、センサデータ系列に対して適切なコンテキストを判断する。本稿では、センサデータ系列に対して手動でコンテキストを付与するツール(SATSUKI Debugger)の実装と、大学構内にて行った実験結果について報告する。

<sup>†</sup> 静岡大学大学院 情報学研究科  
Graduate School of Information, Shizuoka University  
<sup>††</sup> 静岡大学 情報学部  
Faculty of Information, Shizuoka University

## 2. SATSUKI によるセンサアプリケーション開発

### SATSUKI のコンセプト

ここで「センサアプリケーション」とは、環境内や人体に装着したセンサやアクチュエータを用いて、ユーザの要求に応じたタスクを行わせるアプリケーションのことを言う。SATSUKI とは、今後環境内に偏在したり人体に装着したりするであろうセンサやアクチュエータを有効に活用するために、末端のユーザ自身がそれらを用いたアプリケーションを作成、実行できる様にするための枠組みである。センサアプリケーションはその特性上、いたってシンプルなアプリケーションを作成する際にも、環境の状態、センサのスペック等を考慮しなければならず莫大な作成コストがかかることが多い。SATSUKI では、センサのドライバレベルの実装を隠蔽し、ネットワークプログラミングやセンサ間の連携など複雑な処理を代行してくれる枠組みと、アプリケーションを作成するためのインタフェースを備えることで、より容易にセンサアプリケーションの開発ができる環境を提供する。

SATSUKI の狙いは従来の組込型のアプリケーションとは違い、複数のセンサをネットワーク上で統合し、ユーザの必要に応じてその場で即座にアプリケーションを記述することのできる柔軟さであり、アプリケーションは必要になった時点で作成し、役割を果たしたら破棄してしまうという使い方を目的としている。

### SATSUKI の概要

我々はこれまで SATSUKI の枠組みを実現するために、GUI のツールの実装を進めてきた。その GUI ツールの画面を図 1 に示す。SATSUKI は個々のセンサ実体をあらかずコンポーネントと、センサデータを処理するためのコンポーネントをユーザに提供する。各コンポーネントは GUI のツール上で、四角形のオブジェクトとして表示され、ユーザはそれらのオブジェクトをつなぎ合わせることでシステムを記述できる。

SATSUKI のコンポーネントは、ロードされると自身の設定用にウインドウを表示するものが存在する。図 1 の右上のウインドウはコンポーネントをロードしたときに表示されるウインドウで、この場合は「閾値設定コンポーネント」の閾値設定用のウインドウが表示されている。図 1 の左上のウインドウは、SATSUKI のメイン画面である。このウインドウでは COM ポートを開き SATSUKI 環境にセンサを追加したり、「接続設定画面」などその他のウインドウを開く作業ができる。

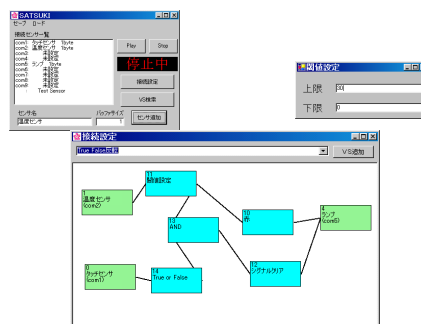


図 1 SATSUKI 画面

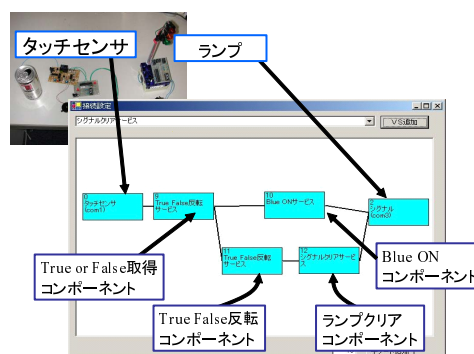


図 2 SATSUKI によるセンサアプリケーション作成例

図 2 に SATSUKI によるセンサアプリケーション作成例を示す。ここではタッチセンサコンポーネントが、センサ実体を表すオブジェクトであり、ランプコンポーネントがアクチュエータを表すオブジェクトである。その他のコンポーネントはセンサデータを処理し、アクチュエータを動作させるための処理用のコンポーネントである。処理用のコンポーネントは接続設定画面の上部にあるコンボックスにより選択し、ロードできる。この例では処理は 2 つに分岐し、タッチセンサの ON によってランプの青を点灯させる処理と、タッチセンサの OFF によりランプを消灯させる処理が実装されている。このように SATSUKI では、ユーザがセンサのドライバレベルの実装や、センサ間連携のための処理を記述することなく、アプリケーションの動作記述のみに注力できるようになっている。

### センサアプリケーション開発の問題点

実際にセンサアプリケーションを実装する際には以下のような制限が存在し、これらがアプリケーションを作成するユーザにとっての高い敷居となっている。環境の制限 環境の制限とは、センサの設置場所における制限である。実際のセンサの取り付けには、壁面の形状、配線の問題、デザイン上 (美観) の問

題、等いくつかの問題が発生することがあり、本当に設置したい場所に設置できない場合が多い。また理想的と思える場所に設置したとしても、本当に理想的なセンサデータが取得されるかどうかは正確には予想できない。

**センサの制限** センサの制限とは、センサの値段やサイズなどを考慮した場合、利用できるセンサに制限があるということである。

**アプリケーション記述の制限** アプリケーション記述の制限とは、どのようなセンサデータが入力されるのか判明しない段階では、アプリケーションの記述を行うことができないということである。どのようなアプリケーションでも入力予測は重要であるが、センサアプリケーションでは入力予測を行うことが困難である。

今回我々が提案する SATSUKI Debugger は、アプリケーションを改良してゆくことで、その環境に「馴染んだ」アプリケーションに仕上げることを支援するものである。環境の制限、センサの制限は非常に多い場合が多く、アプリケーションを改良し続けてゆくことである程度実用に耐える精度を出すことが SATSUKI Debugger の狙いである。

### 3. SATSUKI Debugger

#### SATSUKI Debugger の概要

今まで SATSUKI は、個々のセンサからのデータを用い、それらのフローを記述することでアプリケーションを作成していた。個々のセンサデータを見る方式では前述の「アプリケーション記述の制限」を受け、複雑な状態識別をする際に限界がある。そこで SATSUKI にコンテキスト判断のための仕組みを追加し、ユーザにコンテキストのみを提供し、個々のセンサデータを隠蔽しようというのが今回の提案である。コンテキストを SATSUKI が提供することにより、ユーザは環境やセンサの制限を考慮することなく、アプリケーションの機能面の実装に注力することができる。

しかし近年のコンテキストウェアアプリケーション<sup>7),8)</sup> は、対象物に対する詳細なモデル化を行い、特定環境におけるコンテキストを判断しようとするものが多い。我々が提案する枠組みは、ユーザが目的に応じたアプリケーションを作成するためのものであり、詳細なモデル化を行うことができない。そこで提案システムではモデル化を行わず、実環境の状態を人間が判断し、そのとき取得されたセンサの時系列データにコンテキストを手動で割り振る手法を取る。

SATSUKI Debugger には、センサデータとコンテ

キストを割り当てる「試験段階」と、その情報を元に現実世界で起きているコンテキストを判断し、アプリケーションを動作させる「運用段階」が存在する。そのためコンテキスト定義用とコンテキスト判断用の2つのシステムが必要である。我々は今回 SATSUKI Debugger の実現に向けた第一歩として、センサデータを取得し、コンテキストを定義するためのシステムの実装と検証を行った。今回実装したシステムでは、センサの時系列データとその動作環境を撮影した動画をユーザが見ながら、その動作状態に応じてコンテキストを手動で割り当ててゆく。

#### SATSUKI Debugger の構成

SATSUKI Debugger 全体の構成図を図3に示す。SATSUKI という枠の中にはセンサアプリケーション編集用のツールと、動画、センサデータの取得用のコンポーネントが存在する。動画、センサデータ取得用コンポーネントはアプリケーション編集ツールから呼び出される。実際はアプリケーション編集ツールから呼び出されるコンポーネントはこれ以外にも、各種アクチュエータ制御用コンポーネントや、論理和コンポーネント、論理反転コンポーネント、閾値設定コンポーネント等様々あるが、それらは運用段階で用いるコンポーネントであり、コンテキスト生成までの流れには関係しないので省略した。今回は後のコンテキスト割り当てのために動画データを用いるが、音声データなどを用いてもよく、コンテキストが人間にわかりやすいデータを取得できるものであれば他のセンサを用いてもよいと考えている。

アプリケーション作成の試験段階においてデータ取得用のコンポーネントで取得されたデータは、コンテキスト割り当てシステムに渡される。次にユーザはコンテキスト割り当てシステムを用い、時系列に保存されたセンサデータとコンテキストを対応づける。対応づけられたコンテキストとセンサデータの情報はコンテキスト定義として蓄積される。試験段階では点線で囲まれた「コンテキスト定義システム」の部分が用いられるが、今回実装したのはこの部分である。

アプリケーション作成の運用段階では、コンテキスト判断システムがコンテキスト定義の情報と、実環境から取得されるセンサデータを照合し、コンテキストを選び出す。そしてコンテキストが選ばされると、その情報をアプリケーション編集ツールにフィードバックする。アプリケーション編集ツールはコンテキスト情報を受け取ることができるので、コンテキストに応じたアクチュエータの動作を記述するだけでセンサアプリケーションを構築することが可能となる。

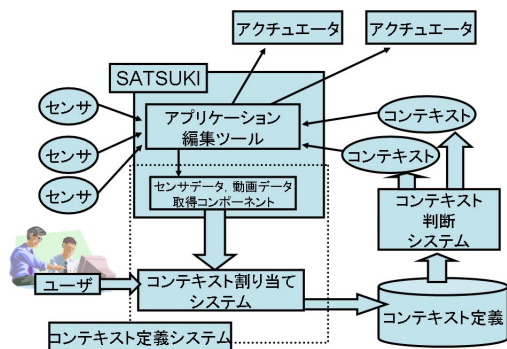


図 3 SATSUKI Debugger 全体図

ここまで試験段階と運用段階と分けて考えてきたが、運用段階に入っても同時にセンサデータと動画データを取得し続けることは可能であり、このように試験段階と運用段階をループさせることで、徐々に求めているシステムに近づけてゆくことができる。これが SATSUKI Debugger をデバッガと呼ぶ所以である。コンテキスト判断の段階で、どうしても判別できないコンテキストの存在があれば、センサ設置場所や種類、数等を変更しなければならないとも考えられるので、そういった場合の検出機能もデバッガとして今後は必須となるだろう。

#### コンテキスト定義システム

今回実装を行ったのがコンテキスト定義システムである、本システムは、センサから取得した時系列データに、ユーザがコンテキストを手動で割り振る作業を支援するシステムである。コンテキスト定義システムは、SATSUKI 上のコンポーネントとして実装したデータ取得用のシステムと、別に用意したセンサデータとコンテキストを対応づけるためのシステムの 2 つから構成されている。本システムではセンサの時系列データを取得すると同時に、USB カメラでセンサの動作環境を録画し保存する。コンテキストの対応付けは USB カメラで撮影された動画を見ながら、ユーザが手動でコンテキストの割り当てを行う。

図 4、6 にそれぞれデータ取得コンポーネント画面、コンテキスト割り当てシステム画面を示す。データ取得コンポーネントは SATSUKI 用のコンポーネントであり、前章で紹介した「閾値設定コンポーネント」同様、ロードされると自身の設定用にウィンドウを表示するタイプのコンポーネントとして実装されている。したがって SATSUKI の接続設定画面上のグラフに追加し、データ取得の条件を記述することが可能になっている。動画取得コンポーネントによって取得された動画例を、図 5 に示す。



図 4 データ取得システム画面図

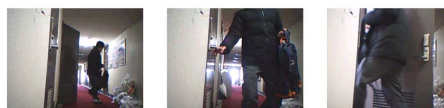


図 5 取得した動画例



図 6 コンテキスト割り当てシステム画面図

コンテキスト割り当てシステムは、データ取得用コンポーネントによって取得されたデータと動画をユーザに表示し、コンテキスト割り当てを行うためのインタフェースツールになっている。コンテキスト割り当てシステム画面の左はコンテキストの分類に応じたツリー表示の画面であり、右の画面はコンテキストを割り振る際に、センサの時系列データと、その際の動画を参照するためのウィンドウである。コンテキスト割り当てシステムの起動時には全てのデータが「Undefined Data」に割り振られている。個々のデータにコンテキストを割り当てるためには、データ名をダブルクリックして、動画を参照するウィンドウを開き、その画面の下部にあるコンボボックスから割り当てたいコンテキスト名を選択する。またコンテキスト名は、ツリー表示ウィンドウの上部にあるボタンをクリックし、そこで表示されるテキスト入力画面によって登録することができる。

#### 4. コンテキスト定義の実験

##### 実験環境

我々はコンテキスト定義システムが正しく動作することと、ユーザがコンテキスト定義を行う際のコスト

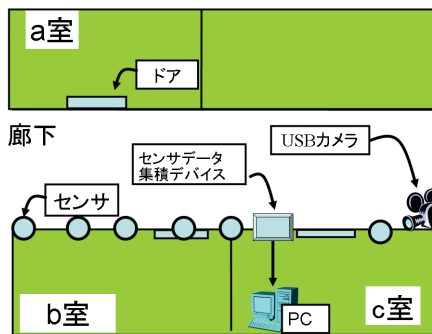


図 7 実験環境

や問題点を探るために、動作実験を行った。

動作実験を行うために、研究室付近の廊下 (2m × 8m) に 6 台の人体感知センサを約 1.5m おきに設置し実験環境を構築した。ここで用いた人体感知センサデバイスは、赤外線量の変化を検知し、人体の移動を検出するもので、人体を検出すると ON の状態になり、非検出状態では OFF になる。実験環境の配置図を図 7 に示し、詳細な実験環境のデータを表 1 に示す。各センサはセンサデータ集積デバイスに有線で接続されており、センサ集積デバイスは各センサの状態変化を取得し、C 室内の PC に送信する。また USB カメラも C 室内の PC に接続されており、動画のデータが取得できるようになっている。

SATSUKI のアプリケーション編集ツール上に、センサデータ、動画取得コンポーネントを配置し、廊下にある人体感知センサと USB カメラからのデータを保存できるようにした。センサデータと動画を取得するタイミングは「TrueFalse カウントコンポーネント」という入力の ON, OFF をカウントするためのコンポーネントを用い、人体感知センサの一つでも ON の状態なら取得を開始し、全てのセンサが OFF の状態に戻ると取得を停止するようにした。図 8 にその際構築した SATSUKI のコンポーネント配置のグラフを示す。

#### 実験内容と結果

C 室内の PC で SATSUKI を動作させ、センサの時系列データとセンサ設置環境の動画データを取得した。データは夜 12 時から翌日の夕方 5 時まで 18 時間の間に渡って取得した。その間 488 パターンのデータが取得できた。

ここで取得したデータの内、午前 8 時から午前 10 時の間に取得されたデータ 60 個に、コンテキスト割り当てシステムを用いてコンテキストを割り当ててみた。また数人の被験者に同様の割り当てをしてもらった。動画を見る時間も含まれるため、60 個程度のコ

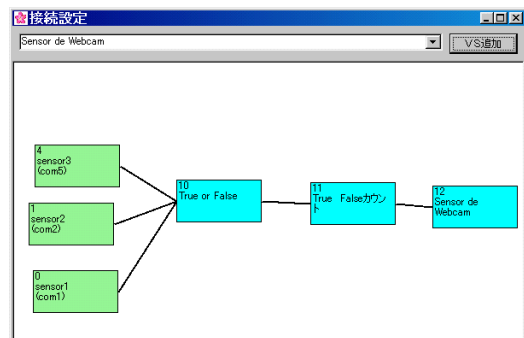


図 8 実験時の SATSUKI 設定画面

表 2 8 時から 10 時までの測定結果例

コンテキスト名	データ数
西側へ通過	10 データ
東側へ通過	26 データ
c 室に入る	1 データ
b 室から c 室へ移動	1 データ
b 室から c 室へ移動	2 データ
a 室から出る	4 データ
c 室から出る	2 データ
二人で西へ	2 データ
不明	12 データ
合計	60 データ

ンテキスト割り当てを行うのに要した時間は 20 ~ 30 分であった。

取得された動画は、廊下を人が歩く映像がほとんどであったが、部屋から顔を出しただけの映像や、何も映っていない映像もあった。作成されたコンテキストは「東側へ移動」、「西側へ移動」、「b 室へ入る」、「c 室へ入る」、「b 室から c 室へ」、「c 室から b 室へ」、「a 室へ入る」という分類がほとんどであった。少数だが「部屋を間違えた」というコンテキストや、「二人で西へ」というコンテキストを割り当てた人もいた。部屋から顔を出したような映像や、人がまったく映っていない映像は「不明」という分類に割り当てられることがほとんどであった。

8 時から 10 時までのデータにコンテキストを割り当てた例を図 2 に示す。

## 5. 考 察

### 取得データ数から得られた知見

今回の実験では 18 時間で 488 個のデータが取得できた。60 個のデータをコンテキストに割り当てるに際して 20 分かかることを考えると、今回取得した全

表 1 実験環境データ

項目	データ
実験に用いた廊下の広さ	幅：約 2m, 長さ：約 8m
センサ間の距離	約 1.5m
用いたセンサ	焦電型赤外線人体感知センサ 6 個
センサデータ集積デバイスに用いたマイコン	PIC16F876
PC スペック	CPU:Pentium4 2.4GHz, メモリ:512M バイト, OS:Windows2000 Professional
USB カメラスペック	Creative WebCam NX Pro(30 万画素)

でのデータにコンテキストを割り当てると三時間程度かかることがわかった。三時間という時間は、多いように思えるが、環境の詳細なモデル化を行うことと比較したら少ない時間とも言える。ただし、一度のコンテキスト割当てでシステムが正しく動作するかどうかはわからないため、今まで十分にデータがとれているコンテキストに際しては自動的にコンテキストを割り当ててしまう、またはデータを取得しないという方法を取り、データ量を削減する仕組みを実装することが望ましいだろう。

大学構内の廊下では、数十名の学生や教員が時間を問わず頻繁に出入りする環境であるため、データがある程度取得できたが、たとえば家の玄関にこのシステムを設置すればデータ数はその十分の一にもならないと思われる。逆に街中など人の混雑する場所においては、人を検出してからその人がいなくなるまで、という今回のようなデータの区切り方では対応できないだろう。こういった場合、一からシステムを作り直すことなく、アルゴリズムの変更によって対処しようとする SATSUKI の利点が発揮できると考えられる。

コンテキスト割当て実験から得られた知見

被験者にコンテキストの割り当てを行ってもらったところ、コンテキストの種類は 6~7 種類程度作られた。出現頻度の少ないコンテキストはデータ数が 2~3 個のものがあり、これではコンテキスト判断の材料としては少ない。また 60 個の内約 20%である 12 個程度がコンテキスト不明という分類に割り当てられていた。これはセンサや USB カメラの設置場所が最適でなかった結果と思われる。これらのことをふまえると、出現頻度の少ないデータの割り当てに人的リソースを割り当てることが重要で、必要のないデータは見なくて済むようにしなければならないことがわかった。

## 6. ま と め

今回は提案システム実現の第一歩として、データ取得とコンポーネントへの割当ての手法を提案し、実装、評価を行った。実験により実際の利用イメージがつかめ、今後の実装に対する方針が得られた。

今後はコンテキスト判断システムの実装を進めると同時に、ユーザに、コンテキスト割当てに重要なデータだけを見せるといった仕組みを考案したい。

謝辞 本論文の執筆に対し、貴重なご意見を頂いた石原進先生、峰野博史先生ならびに、実験に協力頂いた静岡大学情報学部水野研究室、佐藤研究室の皆様にご感謝の意を表す。

## 参 考 文 献

- 1) Patterson, Liao, Fox and Kautz: "Inferring High-Level Behavior from Low-Level Sensors," UbiComp 2003, pp.73-89(2003).
- 2) Essa and Irfan: "Ubiquitous Sensing for Smart and Aware Environments: Technologies towards the building of an Aware Home," Position Paper for the DARPA/NSF/NIST Workshop on Smart Environments (1999).
- 3) 大越, 杉田, 土田, 若山, 西尾, 池田, 徳田: "次世代コンピューティング環境 "smart space" の実現に向けて," 情報処理学会コンピュータシステム・シンポジウム論文集 (2000).
- 4) Starner: "Wearable Computers: No Longer Science Fiction," IEEE Pervasive Computing, pp.86-88 (2002).
- 5) Warwick: "Identity and Privacy Issues Raised by Biomedical Implants," IPTS Report No. 67, EC Joint Research Centre, pp.29-34 (2002).
- 6) 佐々木, 坂根, 太田, 水野: "SATSUKI: 仮想センサコンポーネントの組み合わせによるセンサアプリケーション開発環境" DICO2003, pp.201-204(2003)
- 7) 久住, 北須賀, 中西, 福田, "ウェアラブル/移動情報端末におけるコンテキスト指向プロセス管理", 2002-MBL-23, pp.133-140(2002)
- 8) Hopper, Steggle, Ward and Webster, "The Anatomy of a Context-Aware Application," Proc. of 5th Annual Int 'Conf. Mobile Computing and Networking (Mobicom 99), pp.59-68 (1999)