

業務システムのエンドユーザ開発とフィジカルコンピューティング

End-User Development of Business Information Systems and Physical Computing

古川 慈之*¹
Yoshiyuki Furukawa

澤田 浩之*¹
Hiroyuki Sawada

徳永 仁史*¹
Hitoshi Tokunaga

*¹ 産業技術総合研究所

National Institute of Advanced Industrial Science and Technology

End-user development of software will be one of the key technologies towards the realization of participatory development of business information systems. On the other hand, business information systems consist of not only software but also hardware. In recent years, there is a concept of "physical computing", which means that non-experts make their own computing equipment to interact with real world. This paper reports the development examples which we tried to combine the end-user development with physical computing.

1. はじめに

業務システムの現場参加型開発の実現に向けては、使用者が自ら専用のソフトウェアを作成(エンドユーザ開発)するための技術が重要と考えられる。一方、その業務システムの構成要素として、ソフトウェアだけでなく実世界で動作する機器の存在が重要となる場面は多い。本稿では、そのような機器を非専門家が自ら作成すること(フィジカルコンピューティング)をエンドユーザ開発と組み合わせた開発事例について報告する。

2. エンドユーザ開発の対象範囲拡張

2.1 エンドユーザ開発と MZ Platform

エンドユーザ開発(End-User Development [Sutcliffe 04][Lieberman 06])とは、ソフトウェアおよびシステムの利用者が、自ら開発を実施する形態を指す。一般的な現場参加型開発[西村 13]においては、使用者が自ら開発を実施することまでは含まないと考えられるが、実際にソフトウェアおよびシステムを用いる現場が開発と運用により深く関わると、エンドユーザ開発に近い形態に移行することが見込まれ、そのような状況は業務システムの開発を内製に移行する産業界からの報告事例[日経 11][ICT 14]にも表れている。そのため、エンドユーザ開発で用いられる技術が現場参加型開発に貢献することが期待されている[古川 14]。

MZ Platform[Sawada 04][古川 06]は、エンドユーザ開発のためのソフトウェア開発環境である。MZ Platform を用いることで、Windows や Linux が動作する計算機(PC)上で動作するデスクトップアプリケーションがソースコードを書かずに作成できる。既報[古川 14]では、MZ Platform の概要とそれを用いたエンドユーザ開発事例、およびエンドユーザ開発の課題と展望について述べた。MZ Platform の開発は現在も続いており、エンドユーザ開発の支援機能とその中で用いる技術が向上することで、現場参加型開発に貢献することを目指している。

2.2 MZ Platform を用いた開発対象の範囲拡張

MZ Platform を用いた製造業を対象としたエンドユーザ開発への支援活動が開始から 10 年以上経過し、開始当初と現在で

は社会環境が大きく変化している。エンドユーザ開発支援の活動に大きく影響を与える変化としては、インターネット環境の大幅な普及と情報端末の高性能化および低価格化が挙げられる。

業務システムのハードウェアを構成する計算機は高機能化・低価格化が進行し、数万円程度で高性能な小型 PC や携帯可能なタブレット端末が購入できるようになった。その結果、個人使用のためにこれらを所有している人の数も増加している。このような状況において、MZ Platform のユーザから、Windows や Linux が動作する PC ではなく、スマートフォンやタブレット端末を中心としてシステムを構成するエンドユーザ開発を実施したいとの要望が寄せられるようになってきた。しかし、代表的な iPhone や iPad で用いられる iOS 上では、MZ Platform の動作の前提となる Java が動作しないため、MZ Platform をそれらに対応させることは困難である。また、同様に代表的なスマートフォンやタブレット端末で用いられる Android 上では Java が動作するが、その Java 環境では GUI 用のライブラリが PC と異なるため、やはり MZ Platform を対応させることは難しい。もし大幅な追加開発で対応できたとしても、これらの情報端末上で動作する OS がいつまで使用可能なのか、OS がバージョンアップしても問題なく動作するのか、また今後新たな OS が優勢になった場合に対応させられるのかといった問題が生じることは明らかである。

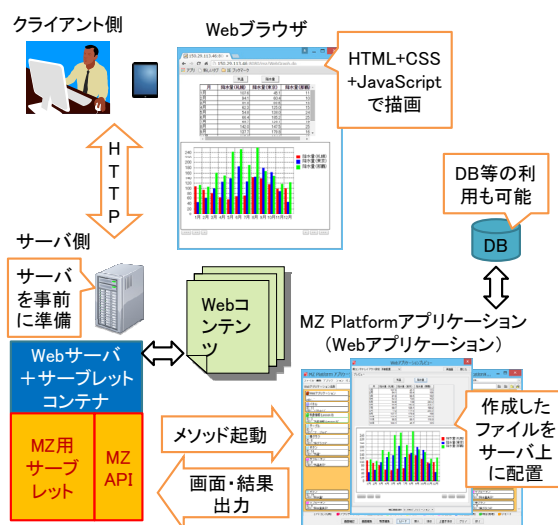


図 1: MZ Platform の Web アプリケーション作成機能

連絡先: 古川慈之, 産業技術総合研究所, 〒305-8564 茨城県つくば市並木 1-2-1, y-furukawa@aist.go.jp

このようなユーザの要望とそれに関する課題を総合的に判断し、MZ Platform では Web アプリケーションを作成するための機能の追加を開始した。ここで Web アプリケーションとは、Web ブラウザを用いて特定の Web サーバにアクセスすることで、所定の処理を実行することが可能なアプリケーション形態を指す。MZ Platform による Web アプリケーション作成機能の概要を図 1 に示す。この機能を用いて作成した Web アプリケーションは、実行するための Web サーバとサーバレットコンテナおよび LAN 環境を別途準備することが必要となる。それらに加えて、MZ Platform 専用のサーバレットをサーバに配備すれば、動作環境の準備が完了となり、その後はユーザが MZ Platform で作成した Web アプリケーションのファイルを Web サーバの所定の位置に配置するだけで、PC のみならずスマートフォンやタブレット端末から Web ブラウザを用いて独自の Web アプリケーションを実行することができる。

当然ながら、Web ブラウザも OS やバージョンによって動作が多少異なるという問題は存在する。しかし、その問題を改善するための Web 関連技術の標準化は現在も進められているため、最新の標準に準拠して MZ Platform の Web アプリケーション作成機能を実現することで、将来的に Web ブラウザごとの差異が吸収される方向に改善されることが見込まれる。さらに、将来新たな OS が動作する端末が優勢になっても、その Web ブラウザが標準に準拠していれば、特に対応せずに MZ Platform の Web アプリケーションが動作することが期待できる。

2.3 フィジカルコンピューティングの位置づけ

スマートフォンやタブレット端末が普及し始めた時期には、フィジカルコンピューティングという言葉も聞かれるようになった。これは、計算機内のソフトウェアに閉じていた計算(コンピューティング)結果を物理的な実世界に反映したり、物理的な実世界の状態に基づいて計算を実施したりすることを指す。それを実現するためのマイコン機器やセンサモジュールが安価で高性能になり、同時に高度な知識を持たなくても実現できるツールキットやソフトウェア開発環境が増えてきたことで、注目されるようになった。

一方、MZ Platform はソフトウェアの開発環境であるが、作成したソフトウェアをユーザが所有している市販の機器と連動して動作させたいという要望は活動当初からあり、主にシリアル通信を用いた使用例が多く存在している。例えば、キーボードではなくバーコードリーダー等の入力機器やデジタルノギス等の計測機器から文字情報を取得して、入力を簡素化する例が挙げられる。つまり、MZ Platform を用いたエンドユーザ開発の対象範囲として、自作のソフトウェアに市販の機器を組み合わせたシステムが既に含まれていたと言える。

近年のフィジカルコンピューティングという概念は、この機器側を非専門家が自作する場合を含んでいる。なお、MZ Platform で当初から支援していた製造業のユーザの中には、市販の PLC とその対応機器を用いてハードウェアを含んだ形でエンドユーザ開発を実施し、工場の自動化を自ら実現するユーザも含まれていたため、そのようなユーザにとってフィジカルコンピューティングの流行はそれほど魅力的ではないと思われる。一方、製造業でも機器の自作になじみがない場合や、小売業やサービス業で簡単な機器を自作できた場合、エンドユーザ開発によってより広い範囲の自動化や業務効率化が実現できることになり、そのような機能に対する潜在的なニーズは広く存在すると考えられる。

つまり、社会環境とユーザニーズの変化に応じて、MZ Platform を用いたエンドユーザ開発の対象範囲を拡大するとい

う対応は必要であり、フィジカルコンピューティングへの対応もその一つと位置付けられる。MZ Platform では、シリアル通信を用いて特定の機器と接続する機能はすでに存在していたが、Web アプリケーションとの組み合わせで新たな使用方法が広がり、かつフィジカルコンピューティングにより自作機器との組み合わせの可能性が広がり、これらによって認識された不足機能の開発を継続して実施しているところである。

3. フィジカルコンピューティングの開発事例

ここでは、MZ Platform を用いたフィジカルコンピューティングの開発事例を 2 つ紹介する。1 つ目は、各種センサからのデータを PC に送信するセンサノードを自作し、それによって計測データを記録するシステムの開発事例であり、2 つ目は、USB カメラの映像から画像処理で検出したイベントを記録するシステムの開発事例である。いずれも、現実世界の状況をデータとしてソフトウェアによる計算の世界に持ち込む開発事例である。なお、これらは著者らが開発した事例であり、実際の MZ Platform ユーザがエンドユーザ開発した事例ではないが、このような開発をユーザが実施しようとした場合の問題点を把握して、不足機能の開発に着手するための先行的な事例と位置付けている。

3.1 事例 1: センサデータと Web 情報の統合と閲覧

図 2 に開発したシステムの概要を示す。各種センサからデータを取得して PC に送信するセンサノード、PC 上でセンサデータを受信して DB に格納する MZ アプリケーション、DB からデータを取得して加工し、Web ブラウザから閲覧できるようにする Web アプリケーションからなる。センサデータを DB に格納する MZ アプリケーションは、並行してインターネット上で公開されている情報(Web 情報)を取得し、同様に DB に格納する。この事例では、DB に MySQL、Web サーバおよびサーバレットコンテナには Tomcat を使用し、すべて単一の PC 上で動作させた。

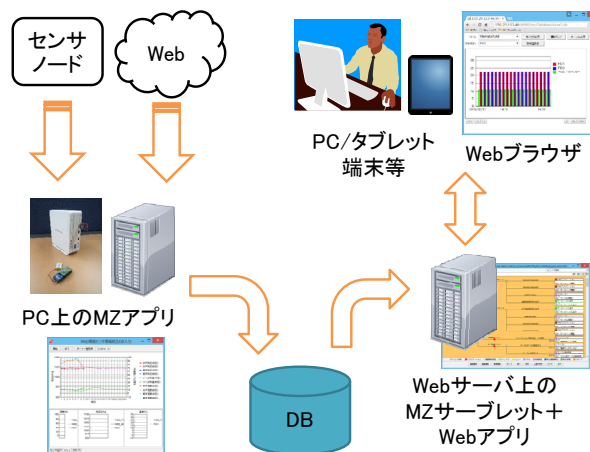


図 2: センサデータと Web 情報の統合と閲覧事例

センサノードの作成には、Arduino FIO を用いた。Arduino FIO から XBee 経由で無線化されたシリアル通信で PC にデータを送信する。PC 側には XBee を USB アダプタで接続する。センサノードで取得するデータは、温度、湿度、気圧の 3 種類である。温度と湿度の取得には SparkFun HIH-6130 デジタル湿度センサ、気圧の取得には SparkFun MPL115A1 搭載 SPI 接続気圧センサを用いた。なお、Arduino の動作指定には専用のソフトウェアでソースコードを記述する必要がある。図 3 にセンサノードの外観を示す。今回作成したセンサノードとしては、リチウ

ムイオン電池を用いたものと、外部から電源を供給するものの 2 種類を準備した。この作成例では、850mAh の容量を持つ電池を使用して、1 分間に 1 度のセンサデータ送信を実施した場合、10 時間程度の連続稼働が可能であった。一方、それ以上の連続稼働が必要となる場合は外部からの電源供給が必要だった。

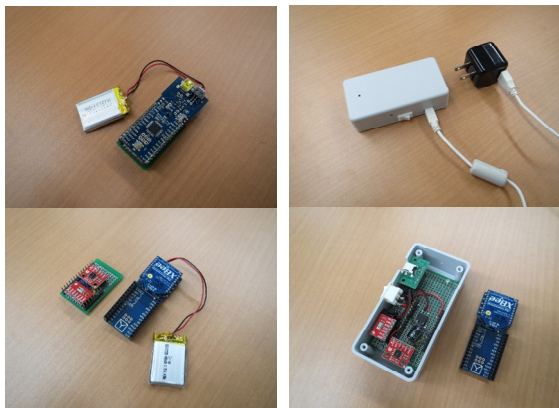


図 3: センサノード外観(電池駆動(左)と外部電源駆動(右))

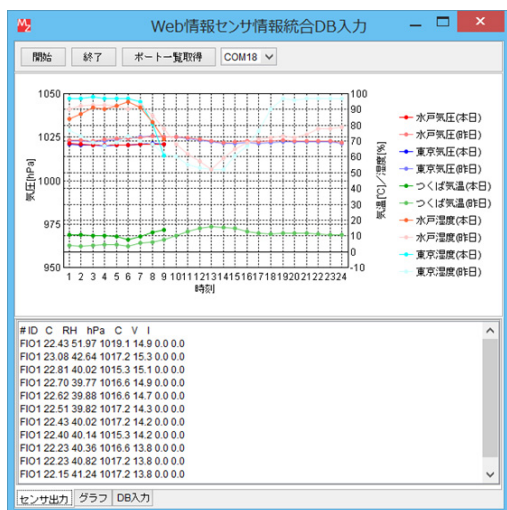


図 4: センサデータと Web 情報の DB 登録アプリケーション

図 4 に PC 上で動作する MZ アプリケーションの画面を示す。PC 側では、XBee 経由の無線化されたシリアル通信で Arduino FIO が出力する固定フォーマットの文字列としてセンサデータが送信される。なお、XBee を適切に設定すると、複数のセンサノードからの送信を単一のシリアルポートで受信することができる。そのため、センサノードからの送信データには、各種センサデータの値に加えてセンサノードの ID 文字列を含めている。受信した MZ アプリケーションでは、ノード ID と各種センサデータに受信した日時情報を加えて、DB に格納する。

この MZ アプリケーションでは、センサデータの他に Web 情報も同様に DB に格納する。今回は、センサデータを取得する温度、湿度、気圧の情報を取得するため、気象庁の Web ページで公開されているアメダスデータを利用した。気象庁の Web ページでは、過去のデータは特定のファイル形式でダウンロード可能となっているが、現在の値については人が閲覧する HTML 形式のデータとして Web ページに埋め込まれている。そのため、この事例では HTML 形式のデータを取得して正規表現を用いて独自に解析し、HTML テーブル内の文字から数値を取得した。センサノードで温度・湿度・気圧を取得した場所は茨城県つくば市であり、温度についてはつくば市の気温を、

湿度と気圧は都道府県庁所在地等のみの提供であったため水戸市と東京 23 区の情報を取得した。これら Web 情報は 1 時間に 1 回、毎時 0 分の情報を 10 分後程度に更新されていたため、MZ アプリケーション側では 10 分間隔で Web 情報を確認し、更新されていたら観測日時を用いて DB に登録する。

DB に格納するテーブルの形式を表 1 に示す。日時・データ取得元・データ種別・値のみで構成される。例として、1 行目はセンサノード「FIO1」から取得した気温の値 22.45 が格納されている。このように、センサノードから取得したデータはデータ種別ごとに別のレコードとして格納される。また、Web 情報も同一の形式で格納され、データ取得元で区別される。

表 1: データベースのテーブル形式と格納例

TIME	SOURCE	TYPE	VALUE
2015/02/12 14:04:20	FIO1	TEMPERATURE	22.45
2015/02/12 14:04:20	FIO1	HUMIDITY	45.26
2015/02/12 14:04:20	FIO1	PRESSURE	998.8
2015/02/12 14:00:00	WEB_ TSUKUBA	TEMPERATURE	11.3
2015/02/12 14:00:00	WEB_ TOKYO	HUMIDITY	34
2015/02/12 14:00:00	WEB_ TOKYO	PRESSURE	1001.9
2015/02/12 14:00:00	WEB_ MITO	PRESSURE	1002.6
2015/02/12 14:00:00	WEB_ MITO	HUMIDITY	48

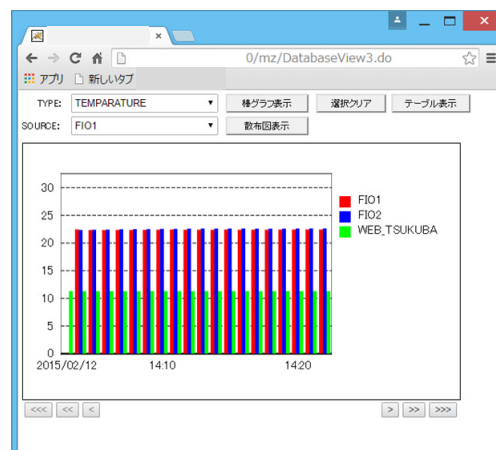


図 5: DB 閲覧 Web アプリケーション

DB に格納したデータを Web ブラウザで閲覧している状態の画面を図 5 に示す。作成した Web アプリケーションでは、DB に格納したレコードをデータ種別ごとにまとめた時系列の棒グラフで各データ取得元の表示する機能と、センサノードのデータ取得元ごとに温度と湿度の関係を表示する機能がある。なお、センサノードからの値はほぼ 1 分間隔で格納されているが、Web 情報からの値は 1 時間間隔で格納されているため、Web アプリケーションで描画する際に Web 情報からの値は毎分の値を 1 時間同じ値で表示するようにしている。

3.2 事例 2: USB カメラ映像からの人検出記録と閲覧

図 6 に開発したシステムの概要を示す。PC 上で USB カメラの映像から画像処理で人を自動検出し、検出日時とその画像を DB に格納する MZ アプリケーションと、DB からデータを取得して加工し、Web ブラウザからデータと画像を閲覧できる Web アプリケーションからなる。

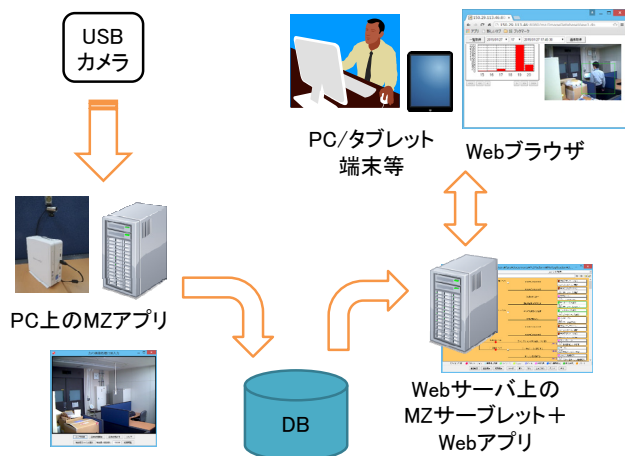


図 6: USB カメラ映像からの人検出記録と閲覧事例

USB カメラから映像を表示して画像を取得する機能は MZ Platform の標準機能としてすでに存在する。取得した画像から自動で人を検出する画像処理は、OpenCV を用いて実現した。使用した OpenCV のバージョンは 2.4.6 である。このバージョンでは Java API が提供されているため、機能呼び出しのための MZ Platform コンポーネントを作成し、MZ アプリケーションから実行した。呼び出した機能は、機械学習結果の XML ファイルを指定して画像内の物体検出を実行する機能で、OpenCV の配布に含まれる歩行者を検出するための XML ファイル (hogcascade_pedestrians.xml) を使用した。実行の様子を図 7 に示す。人を検出した際には、その日時と検出領域を矩形で囲んだ検出結果画像を DB に格納する。なお、使用した USB カメラの解像度は 640×480 であったが、OpenCV による画像処理時に解像度を 320×240 に縮小して処理を行っている。そのため、DB に格納する画像も解像度は 320×240 となっている。



図 7: OpenCV による人検出と DB 登録アプリケーション

DB に格納したデータを Web ブラウザで閲覧している状態の画面を図 8 に示す。作成した Web アプリケーションでは、DB に格納されたデータから年月日を選択し、続いて時刻を選択する。それぞれ選択する際に格納されたデータ数がどの程度かをグラフとして表示する。最終的に日時を指定した検出結果画像が画面に表示される。

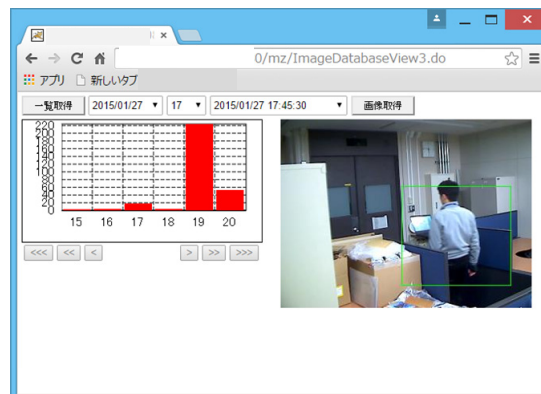


図 8: DB 画像閲覧 Web アプリケーション

3.3 結果と考察

2 つの開発事例について、MZ Platform を用いて目的とする機能が実現できたため、このような事例を今後ユーザが独自に開発することは可能であると言える。一方、両者で共通する DB からデータを取得して加工し、Web ブラウザでグラフや画像を閲覧する機能については、実行速度や安定性の観点から今後も MZ Platform 側の機能開発を継続する必要があることがわかった。また、センサノードの自作については、必要な情報が入手しやすくなっているが、ソフトウェアの自作よりさらにノウハウが必要となることがわかったため、エンドユーザ開発とフィジカルコンピューティングとの組み合わせについては必要な機能と支援方法について今後検討する予定である。

参考文献

- [Sutcliffe 04] Sutcliffe, A., Mehandjiev, N.: End-User Development, CACM, 47(9), pp.31-32, 2004.
- [Lieberman 06] Lieberman, H., et al.(ed.): End-User Development, Springer, pp.31-32, 2004.
- [西村 13] 西村ら: 特集「介護・医療システムの現場参加型開発」にあたって, 人工知能学会誌, 28(6), p.879, 2013.
- [日経 11] 日経コンピュータ編: 開発・改良の切り札 システム内製を極める, 日経 BP, 2011.
- [ICT 14] ICT 経営パートナーズ協会: 超高速開発が企業システムに革命を起こす, 日経 BP, 2014.
- [古川 14] 古川ら: 業務システムを対象としたエンドユーザ開発支援の課題と展望, 2014 年度人工知能学会全国大会 (第 28 回) 論文集, 1L4-NFC-05a-3in, 2014.
- [Sawada 04] Sawada, H., et al., Y.: A Manufacturing Software Development and Operation Framework "MZ Platform" and its Applications in Industry, In Proc. of Advanced Engineering Design, 2004.
- [古川 06] 古川ら: MZ Platform イベント駆動型コンポーネント指向開発環境を用いたエンドユーザ開発への試み, 情報学会第 68 回全国大会講演論文集(1), pp.199-200, 2006.