

National Institute of Advanced Industrial Science and Technology
AIST

エンドユーザ開発による企業内情報化への取り組み

○古川慈之 澤田浩之 徳永仁史 手嶋吉法 松木則夫
産業技術総合研究所 デジタルものづくり研究センター

2010年3月4日 第9回SIG-KST発表資料 1

National Institute of Advanced Industrial Science and Technology
AIST

背景: 中小製造業の情報化

- 企業の要求1: 業務効率化・生産性向上
 - 直接業務の情報化(業務用ソフトウェア開発)
 - 開発コストとリソースは限定、ソフトウェアは各業務固有
 - ソフトウェアの使用者(エンドユーザ)による開発と運用が必要
 - 例: 技術情報管理、工程設計・管理、生産管理、品質管理、...
 - 「ものづくり・IT融合化推進技術の研究開発」(NEDOプロジェクト)の一部として
2001年から“MZ Platform”の研究開発を開始

2010年3月4日 第9回SIG-KST発表資料 2

National Institute of Advanced Industrial Science and Technology
AIST

目標

- エンドユーザ開発で企業内情報化を促進する
 - 業務効率化・生産性向上で企業の競争力強化
 - 業務プロセスの形式知化と企業内情報の蓄積・可視化で知識・技術を継承

<エンドユーザ開発とは>

- Sutcliffe, A., Mehandjiev, N., "End-User Development", Communications of the ACM, 47(9), pp.31-32, 2004.
- 必要な機能(ソフトウェア)を自ら実現して使用する
※高度な開発スキルを要せずに
→ここでは外注せずに内製で業務用ソフトウェア開発をすることを指す

2010年3月4日 第9回SIG-KST発表資料 3

National Institute of Advanced Industrial Science and Technology
AIST

補足: エンドユーザ開発(EUD)

EUDツールによる学習コストと適用範囲のトレードオフ
Fischer, G., et al., Meta-Design: A Manifesto for End-User Development, Communications of the ACM, 47(9), pp.33-37, 2004. の図を修正

2010年3月4日 第9回SIG-KST発表資料 4

National Institute of Advanced Industrial Science and Technology
AIST

産総研による取り組み

- ソフトウェア開発環境MZ Platformの研究開発
- MZ Platformの普及活動: 企業への導入を推進
- MZ Platformを基にしたソフトウェア自動合成の研究開発

2010年3月4日 第9回SIG-KST発表資料 5

National Institute of Advanced Industrial Science and Technology
AIST

MZ Platform (2001~)

- コンポーネント指向ソフトウェア開発環境
 - 2004年より配布開始(研究会登録+会費制)、無料講習あり
 - 動作環境: Windows+Java (+Java3D)
 - 個人・法人会員数 約450 (ライセンス発行数 約1500)
 - ソフトウェア開発企業へのライセンス移転契約10件
 - 詳細は研究会HPへ

MZ Platform提供機能

2010年3月4日 第9回SIG-KST発表資料 6

*)本資料の著作権は著者に帰属します。

補足:コンポーネント指向開発

- コンポーネント指向開発:エンドユーザ開発を実現するための技術の一つ
 - Morch, A.L., Stevens, G., Won, M., Klann, M., Dittrich, Y., Wulf, V., "Component-Based Technologies for End-User Development", Communications of The ACM, 47(9), pp.59-62, 2004.
- "コンポーネント"の組み合わせに基づくソフトウェア開発
 - コンポーネントとは
 - 再利用可能なソフトウェア部品
 - ある程度まとまった処理の単位
 - コンポーネントの再利用による開発コストの低減
 - コンポーネント間の接続を記述すればよい
 - ユーザによる実現が難しい機能のコンポーネント化により、エンドユーザ開発の適用範囲が引き上げられる

2010年3月4日 第9回SIG-KST発表資料 7

MZ Platformにおける記述方式

- コンポーネント配置+イベント駆動の処理記述
 - コンポーネント配置はインスタンス生成、リフレクションで動的結合
 - 記述しながら実行可能で実行中に修正可能
 - マウス操作と少数のキー入力でコンポーネント配置・接続
 - 簡単にアプリケーション作成⇒ソースコードを書かなくていい

2010年3月4日 第9回SIG-KST発表資料 8

MZ Platformのアプリケーション作成画面

- アプリケーションビルダー:コンポーネント間の接続を定義してアプリケーションを作成

作成したアプリケーション例

2010年3月4日 第9回SIG-KST発表資料 9

比較:Javaによる実現例

- 同様のアプリケーションをJavaのソースコードで記述した場合

作成したアプリケーション例

2010年3月4日 第9回SIG-KST発表資料 10

解決すべき課題:コンポーネントの粒度と再利用

- コンポーネント指向開発のジレンマ
 - 再利用による効果はなかなか上がらない

非依存性: 再利用性が高い	高機能: 再利用時の効果は大
低機能: 再利用時の効果小・ 接続のオーバーヘッドが大	依存性: 内部データ・ロジックの 再利用性が低い傾向

小 ← コンポーネントの粒度 → 大

カスタマイズするためにソースレベルでの編集が必要

2010年3月4日 第9回SIG-KST発表資料 11

解決へのアプローチ

- カスタマイズ性と再利用性の両立を目指して
 - MZ Platformではコンポーネント間の接続はソースコーディングなし⇒修正が容易+接続のオーバーヘッド小
 - アプリケーション特有のロジックとデータ構造はアプリケーション層に
 - 多様な処理がソースコーディングなしで実現できるように

2010年3月4日 第9回SIG-KST発表資料 12

National Institute of Advanced Industrial Science and Technology
AIST

細粒度コンポーネントの整備とユーザ定義データ型の導入

- 細粒度コンポーネントの整備
 - データ処理・ロジック記述の最小単位をコンポーネント化
 - 汎用データ処理
 - 文字列・数値・日付・リスト・テーブルetc.
 - 値の取得と設定、基本的な操作
 - 処理の最小単位
 - 分岐・繰り返し・判定・演算etc.
- ユーザ定義データ型の導入
 - 細粒度コンポーネントとして散在するデータを一括管理するためのデータ構造・コンポーネントの導入
 - データに名前をつけて格納
 - データの名前でアクセス

ユーザ定義データ型用リスト構造

- 名前 (文字列)
- 要素 (オブジェクト)
- 型名 (文字列, オプション)

汎用データ型
同リスト構造
一階層化

2010年3月4日 第9回SIG-KST発表資料 13

National Institute of Advanced Industrial Science and Technology
AIST

ユーザ定義型コンポーネントの実現

- コンポーネントの組み合わせで複合コンポーネントを定義
 - アプリケーションの階層化
 - 処理やデータの隠蔽: 特定のメソッド公開とイベント伝播
 - 細粒度コンポーネント+ユーザ定義データ型
= ユーザ定義型コンポーネント

複合コンポーネントの例

2010年3月4日 第9回SIG-KST発表資料 14

National Institute of Advanced Industrial Science and Technology
AIST

MZ Platformの技術的なポイントのまとめ

- コンポーネント(インスタンス)の結合でアプリケーションを構築
- イベント処理の共通コネクタを定義
- リフレクションによる動的結合
- 細粒度コンポーネントの整備とコンポーネントの階層化でカスタマイズ性と再利用性を両立

2010年3月4日 第9回SIG-KST発表資料 15

National Institute of Advanced Industrial Science and Technology
AIST

MZ Platformの活用場面

- ◎ 各企業に特化したソフトの開発
工程・生産管理など、各企業の業務形態に大きく依存したソフトの開発。
- ◎ 既存データの有効活用
データベースアクセスコンポーネントによる、EXCEL、ACCESS、その他SQL系データベースに対するデータの読み書き。
- 3次元CADデータの利用
形状モデル管理&3Dビューワコンポーネントによる、3次元CADデータ(IGES/STEP/DXF)の読み込み、表示、参照。
- △ 特殊処理の開発
有限要素法による構造解析処理機能など、特殊な処理の開発には不適。
- △ 大規模データ処理機能の開発
銀行の口座振替や列車の座席予約のように、大量データ処理と即時性が要求されるソフトウェアシステムの開発は、従来方法が適切。

2010年3月4日 第9回SIG-KST発表資料 16

National Institute of Advanced Industrial Science and Technology
AIST

生産管理システムの開発で利用される主なコンポーネント

- データベースアクセス
 - 外部データベースとの通信(データの登録、更新、検索)
 - 通信確認済みデータベース
MS-Excel、MS-Access、SQL Server (MSDE)、Oracle、MySQL、PostgreSQL
- テーブル、グラフ
 - 表データ(データベース検索結果等)の表示、編集
- 帳票
 - 伝票、バーコードの発行

2010年3月4日 第9回SIG-KST発表資料 17

National Institute of Advanced Industrial Science and Technology
AIST

データベースアクセスとデータ表示

テーブル

積上げ棒グラフ

データベースアクセスコンポーネント

ガントチャート

MS-Excel, MS-Access, SQL Server, Oracle, MySQLなど

テーブル テーブル テーブル

データベース

(参考) 工程管理システム開発版

2010年3月4日 第9回SIG-KST発表資料 18

伝票の作成・発行

バーコードやQRコード、表、画像の貼りこみ

2010年3月4日 第9回SIG-KST発表資料 19

バーコード入力

バーコードリーダからの入力読み込み

製品名とロット番号を表示

2010年3月4日 第9回SIG-KST発表資料 20

MZ Platformによるエンドユーザ開発の得失

- ソースコードを書かずにソフトウェアを作成できる
 - ソフトウェア開発経験がなくても比較的容易に作成できる
 - 実際に作成するにはそれなりに学習コストがかかる
- カスタマイズ可能なコンポーネントの再利用で効率的
 - 作成できなくても修正できる場合もある
- ソフトウェアに対する要求仕様を考える力が必要
 - 業務の流れ、情報を整理する必要あり
 - すべてを一度に情報化するのは困難なので取捨選択が必要

→完全なエンドユーザ開発ではなく、支援体制が必要

2010年3月4日 第9回SIG-KST発表資料 21

**MZ Platform普及体制モデル
—ケーススタディとビジネス展開—**

2010年3月4日 第9回SIG-KST発表資料 22

システム開発事例1: 作業実績収集システム
—プラスチック射出成形企業(大分県)—

資料提供: 大分県産業科学技術センター

低コスト・短期間でのシステム自社開発

- MZプラットフォーム(システム開発ツール)
- エクセル
- MSDE(無償データベースソフト)
- PC、バーコードリーダ、ハンディターミナル

手作業による書き直しから
電子データによるオンライン化へ

時間単位の進捗・数量管理
→ 納期・在庫管理精度の向上!
手書き作業の排除
→ 記入ミス排除、作業時間20%削減!

半日単位の進捗・数量管理
手書き作業 平成19年度九州経済産業局IT経営力大賞特別賞受賞

2010年3月4日 第9回SIG-KST発表資料 23

システム開発事例2: 受注・外注・進捗管理システム
—金型製造企業(佐賀県)—

開発支援体制

- 佐賀県工業技術センター、ソフトウェアベンダーによるサポート
- 産総研(茨城)での技術研修1ヶ月

システム構成要素

- MZプラットフォーム(システム開発ツール)
- MySQL(無償データベース)

パッケージソフトから
自社開発システムへ

加工時間の正確な把握による原価管理の高精度化

既存システム(パッケージソフト)

- 不正確なデータ登録(日報の手入力)
- 低い処理能力(システムの老朽化)
- 困難な機能改善(外注費の発生)

全社員によるリアルタイムの情報共有
→ 原価意識・利益意識の向上!
システムに対する現場要望の即時反映
→ 機能改善・活用意欲の向上!

平成20年度九州経済産業局IT経営力大賞特別賞受賞

2010年3月4日 第9回SIG-KST発表資料 24

システム開発事例3: 受注・製造・在庫管理システム — 金属表面処理企業(長崎県) —

開発体制
第1ステージ: 地元ソフトベンダーとの共同作業
第2ステージ: 自社開発

システム構成要素

- MZプラットフォーム(システム開発ツール)
- Microsoft エクセル
- Microsoft SQL Server

手作業による書き写しから
電子データによる一元管理へ

受注票 出荷票 計画表 製品・部品台帳

生産性の正確な把握と分析!

2010年3月4日 第9回SIG-KST発表資料 25

システム開発事例3 業務分析図

生産管理 業務フロー

2010年3月4日 第9回SIG-KST発表資料 26

システム開発事例3 システム構成

管理部門 製造部門

受注登録 部品生産計画 製品生産計画 出荷登録 エクセル入出力(Visual Basic) 部品生産 製品生産

2010年3月4日 第9回SIG-KST発表資料 27

システム開発事例3 業務フローとシステム入出力

2010年3月4日 第9回SIG-KST発表資料 28

アプリケーション開発による実証評価結果

操作方法習得期間: 従来(MFC/C++, Java)の半分以下
アプリケーション開発工数:

実施企業	開発内容	開発工数	従来との工数との比較
企業組合(長野)	企業間工程管理	30人日	従来1/4以下
切削加工(大阪)	技術情報活用	3人日	従来1/10以下
	品質検査	10人日	従来1/3以下
板金加工(長野)	工程設計支援	7人日	従来1/4以下
プレス加工(長野)	生産・帳票管理	30人日	従来1/10以下
プラスチック射出成形(大分)	作業実績収集	45人日	従来1/3以下
射出成形金型(東京)	日程・進捗管理	30人日	従来1/4以下
研磨加工(福岡)	受注・工程・品質管理	25人日	従来1/3以下

従来工数との比較は、同機能のアプリケーションを従来の開発環境で構築するとした場合に必要工数を、ソフトウェアベンダーの技術者に概算してもらったことによる。

2010年3月4日 第9回SIG-KST発表資料 29

エンドユーザ開発で解決すべき課題

- プログラム作成以外にも必要な作業が多い
= 作成と利用へのハードルが高い
- 業務分析
- アプリケーション設計
- 導入・保守

2010年3月4日 第9回SIG-KST発表資料 30

モデル記述によるソフトウェア合成(2006～)

- 業務プロセスを表現する最小限の入力から実行可能なソフトウェアを自動合成する
 - 業務プロセスのモデルをアプリケーションの仕様としてアプリケーション設計を支援する
 - エンドユーザ開発への貢献

2010年3月4日 第9回SIG-KST発表資料 31

対象範囲とアプローチ

- 関連キーワード
 - モデル駆動開発(MDD)
 - ドメイン固有言語(DSL)
- 対象:個人で完結する(とみなせる)業務プロセス
 - 業務に関連する情報を作成・処理・解析するプロセス
 - ソフトウェアとして実現可能な段階までプロセスを詳細化
 - 業務プロセスをソフトウェアによる処理のフローとしてモデル化
 - 対象とするドメインを限定することでフローに対応するモデルの記述を簡略化
- 対象外:協調作業とワークフロー管理
 - 関連技術の例: BPMNで業務プロセスを記述してBPEL4WS形式に変換してワークフロー管理システムで実行する

2010年3月4日 第9回SIG-KST発表資料 32

ソフトウェア合成のシステム構成

MZ Platformの機能であるアプリケーションの動的生成を基に、モデル記述からのソフトウェア合成を実現。モデル記述や自動合成のツール及び使用する部品はすべてMZ Platform上のソフトウェアとして構築。

2010年3月4日 第9回SIG-KST発表資料 33

ソフトウェア合成の例: 製造業向け計算ツール

2010年3月4日 第9回SIG-KST発表資料 34

ソフトウェア合成のまとめ

- 必要なソフトウェアの動きをモデルとして記述するとMZ Platformアプリケーションが自動的に合成される
 - アプリケーション設計が完了してからコンポーネントの配置と接続の作業を自動化
 - ソフトウェアの処理のプロセスを詳細に記述する必要あり
 - 手動で作成する場合と比較して対象範囲を限定することで実現

2010年3月4日 第9回SIG-KST発表資料 35

MZ Platformによるエンドユーザ開発の今後の展開の可能性

- コンポーネントを追加してアプリケーションの対象分野を拡張
 - データ管理系だけではなくCAD系や計測・制御系など
 - 他のツール、ミドルウェアとの接続など
- コンポーネントの機能を改善して開発をより簡単に
- 自動合成機能を拡張して開発をより簡単に
- 別言語や別形態で実装してアプリケーションの適用範囲を拡張
 - Javaによるデスクトップアプリケーションだけではなく、.NETやJavaScriptでの実装、Webアプリケーションへの適用など

2010年3月4日 第9回SIG-KST発表資料 36