



ADempiere とは...



自己紹介



一言

「オープンソースのERPと、千葉ニュータウンが発展すれば良いなと思っています。よろしくお願い致します。」

印西市のゆるキャラ
インザイ君⇒



OSS-ERP歴
約4年半

名前	はぎわら ひであき
	萩原 秀明
年齢	38歳(2児の父)
住所	千葉県印西市 (千葉ニュータウン)在住
URL	compiere-distribution-lab

略歴

- ◆学生時代は筋トレと会計の勉強に励む!!
- ◆大手商用ERPの導入・開発に4年ほど携わる(主に会計系)。

約6年



この間には色々ある。

天国

モテ期

結婚

長女
誕生
転職

苦悩

挫折

- ◆OSS-ERP Compiereの導入・開発に3年半ほど携わる。
- ◆現在は、株式会社オープンソース・ビジネスアプリケーションズで業務アプリケーションエンジニアとして活躍中!!

ADempiere紹介 -目次-



ADempiereとは



業務機能概要



特徴的な機能紹介



開発手法紹介



ADempiere活用のポイント

無料で使用できるオープンソースのERP [GPL(General Public License)]

-  購買管理、在庫管理、販売管理、生産管理、会計管理、顧客管理など、一連の業務機能が既に実装されている、無料で使用できるオープンソースのERP。
-  中小企業・大企業の子会社などを主なターゲットとした、無料で使用できるオープンソースのERP。
-  1企業で導入できるのはもちろん、グループ企業(複数企業)にも導入できる、無料で使用できるオープンソースのERP。
-  世界中で使用されておりグローバル対応されている、無料で使用できるオープンソースのERP。
-  クラウド環境でも利用できる、無料で使用できるオープンソースのERP。
-  Compiere ⇒ ADempiere ⇒ iDempiere (OSGi + ADempiere)と発展している無料で使用できるオープンソースのERP。

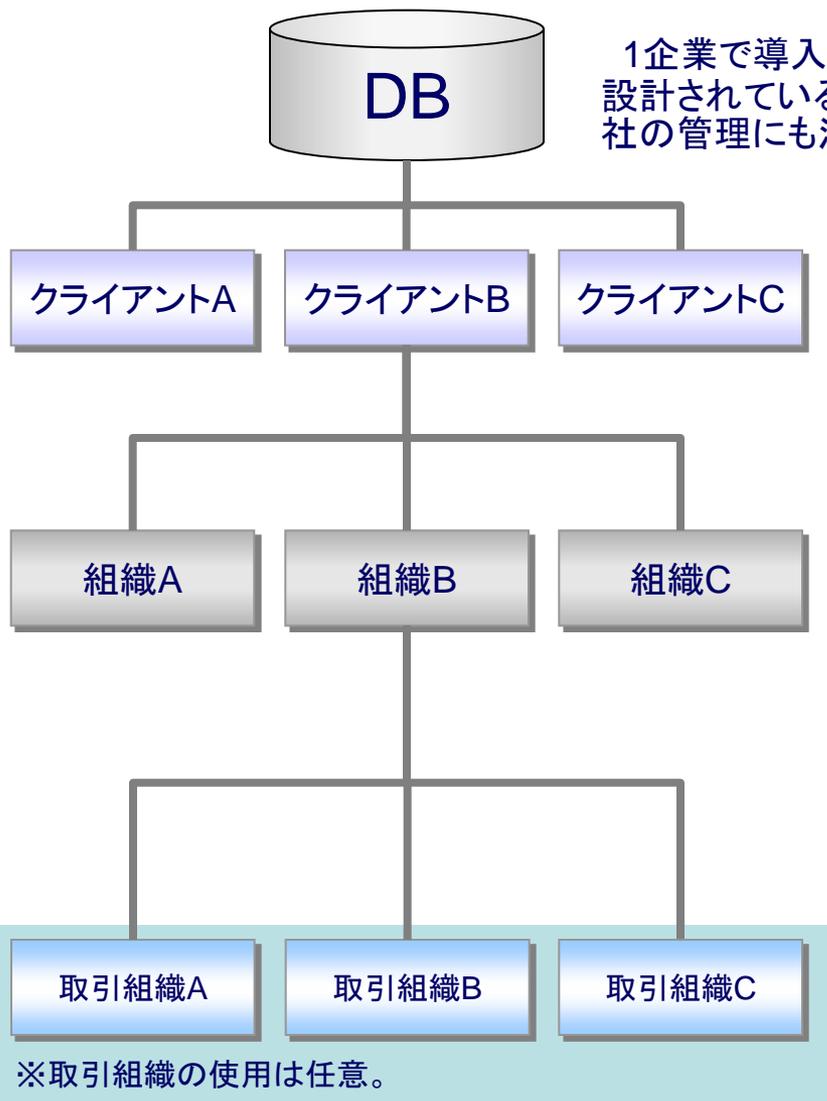
業務アプリケーションの開発基盤

-  オープンソースとして世界中の人々が開発に参加するのは、有用性がありシステムアーキテクチャが優れている証拠。足りない機能があっても開発し易いシステムアーキテクチャになっているので比較的容易に開発できる。 (開発言語: Java / DB: Oracle、PostgreSQL)

業務機能概要 (ADempiere)



特徴的な機能: 単体企業にもグループ企業(複数企業)にも導入可能



1企業で導入できるのはもちろん、グループ企業(複数企業)にも導入できるように設計されている。グループ内の子会社だけでなく、フランチャイズ企業や販売会社の管理にも活用できる。

クライアント[Client] = 会社 or 企業グループ

- 1つのDB上に複数のクライアントを作成する事ができる。
(マルチクライアント)。
- クライアント間ではマスタの共有やトランザクションデータの参照は基本的にできない。(クライアント間の独立性が高い)

組織 [Organization] = 部署 (事業部) or 会社

- グループ化された階層化構造を複数作成できる。
- 組織間でマスタを共有できる。
- 組織間取引ができる。
 - カウンター伝票
⇒(例)社内取引で組織Aで発注伝票を起票すると組織Bで対応する受注伝票が作成される。
 - 本支店会計の仕訳も起票する事ができる。

取引組織 [Transaction Organization] = 部署(事業部)

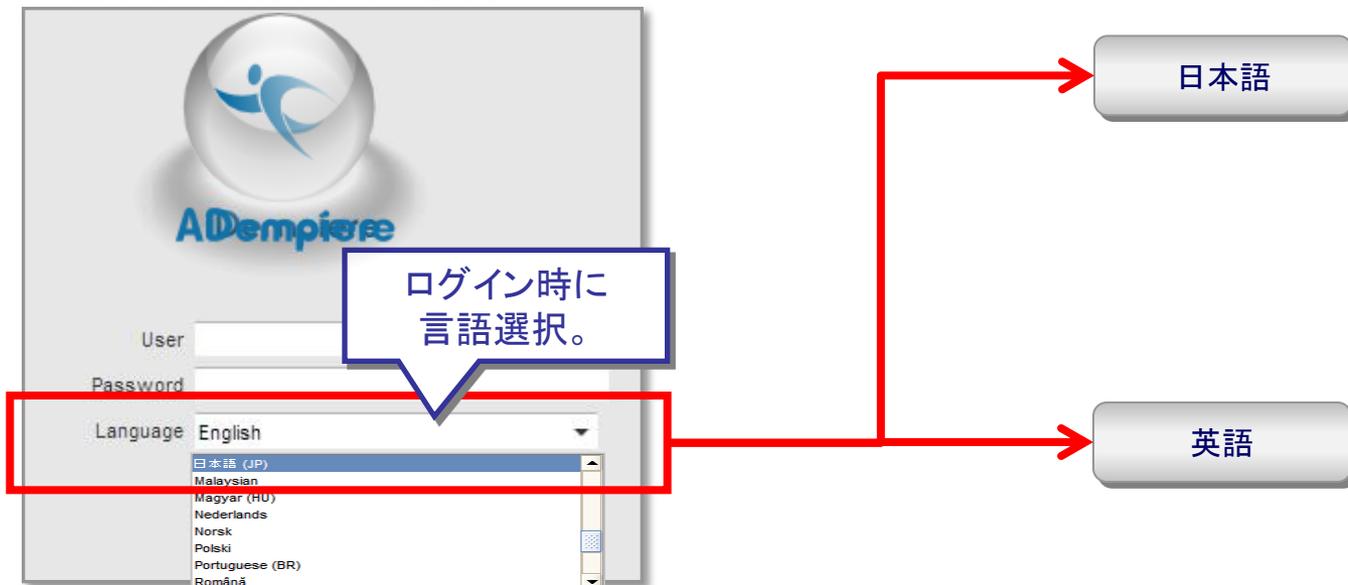
- クライアントを1つの企業グループと見立て、組織を1つの会社と定義した場合、取引組織を使用して社内の子会社データを管理する事ができる。

特徴的な機能:グローバル対応



多言語対応(マルチ言語)

- XML形式の翻訳ファイルがあれば、色々な言語でUIを表示する事ができる。
- システム上で翻訳作業を行う事もできる。



外貨対応(多通貨対応)

- 世界中の通貨が登録済み。レート登録を行えば、すぐに外貨取引が行える。
- 複数のレート登録が可能(標準機能でスポットレート/平均レート/期末レート/会社レートが用意されている)。
- 仕訳データには会計通貨(機能通貨)/取引通貨の2種類の通貨情報を保持。

※会計通貨(機能通貨)は会計基準毎に設定が可能。

特徴的な機能: 会計



複数の会計基準への対応(複数の会計帳簿を作成できる)

日本の現状の会計帳簿



1つの会計帳簿が3つの役割を担っている。

- 財務会計用帳簿
- + 税会計用帳簿
- + 管理会計用帳簿



国際会計基準用の会計帳簿



大企業グループでは、日本国内の財務会計基準の帳簿に加え、国際会計基準の会計帳簿も必要になってくる。



海外(米国)の会計帳簿

役割に応じて、会計帳簿が別れている。

- 財務会計用帳簿
- 税会計用帳簿
- 管理会計用帳簿



柔軟性のある消費税処理

ADempiereの消費税処理は、世界中の消費税に類似する税金(付加価値税や営業税など)も考慮して設計されており、柔軟性が高い。

- 品目別消費税設定
 - 取引地域別消費税設定
 - 国税/地方税別仕訳設定
 - 伝票明細単位での消費税計算 ←→ 伝票単位での消費税計算
 - 営業税として費用処理
- など...



多次元分析レポート

ADempiereでは管理会計の分析軸として、あらかじめ下記のような分析軸が設定できるようになっており、さらに必要に応じて、ユーザー定義として分析軸をカスタマイズで追加する事ができる。

組織 [AD_Org_ID]	取引組織 [AD_OrgTrx_ID]	品目 [M_Product_ID]	取引先 [C_Bpartner_ID]	販売地域 [C_SalesRegion_ID]
プロジェクト [C_Project_ID]	キャンペーン [C_Campaign_ID]	アクティビティ [C_Activity_ID]	ユーザー 定義1	ユーザー 定義2

特徴的な機能：内部統制機能



権限管理

➤ ADempiereの権限は“職責[Role]”で管理され、職責をユーザーに割り当てる仕組みになっている。職責には業務メニューやデータのアクセスが許可される“組織[Organization]”などを設定するようになっている。

(例)



※“職責(Role) + ユーザー”をキーとしてアクセスできる組織を制御する事もできる。



ログ管理

常にログ管理の対象

“変更履歴”を記録するかどうかは職責やテーブル毎に制御できる。

全てのレコードのデータ登録者/最新更新者

ログイン
(セッション)履歴

プロセス履歴
(実行パラメータ/ログ)

変更履歴

職責 変更ログ

テーブル 変更ログ

特徴的な機能:データのエクспортとインポート



データのエクспорт

➤ 一覧レポートより下記のファイル形式でエクспортする事ができる。

Postscript

XML

PDF

HTML

タブ区切り

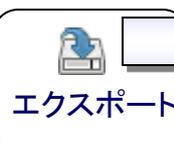
セミコロン区切り

csvカンマ区切り

xls

伝票 No	注文書	注文日付	顧客注文番号	説	明
100000	80000 - 2003-11-01 00:00:00	2003/11/01			Export Order
200000	80000 - 2002-02-22 00:00:00	2002/02/22		(1)	
200001	80001 - 2002-02-22 00:00:00	2002/02/22		(2)	
200002	80002 - 2002-08-01 00:00:00	2002/08/01	Phone Order		Text is printed in the language of the business partner (if not entered in English like this). This is also a text, which will be printed in block format on two lines on the documents.

Page 1(1,1) of 6(1,6)



保存先: My Documents

- AS3_project
- CompiereDB仕様書
- Compiere自作資料
- DVD
- D&PASS
- My GOLD
- My Music
- My Pictures
- My Videos
- My Webs
- NetBeansProjects
- seminar
- Symantec
- ダウンロード
- ネットワークツール
- 作業フォルダ_ADempiere
- 作業フォ
- 個人用図

ファイル名: []

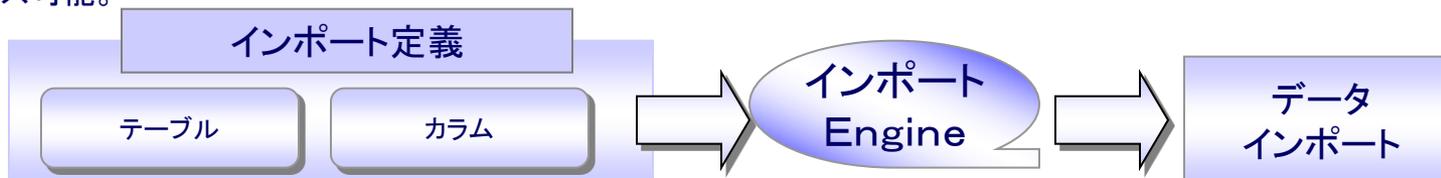
ファイルタイプ: xls - Excel file

- ps - Postscriptファイル
- xml - XML ファイル
- pdf - Acrobat PDFファイル
- html - HTMLファイル
- txt - タブ区切りファイル
- ssv - セミコロンで区切るファイル
- csv - カンマで区切り Excelファイル
- xls - Excel file



データのインポート

製品マスタ/取引先マスタ/勘定科目/在庫データ/注文書/請求書/仕訳データ/支払データなどのデータインポートツールが用意され、カスタマイズ可能。



特徴的な機能: Web Service

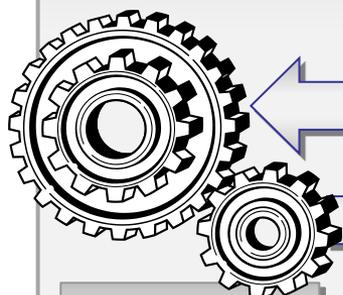


Web Service

Webサービス(ウェブサービス)とは、HTTPなどのインターネット関連技術を応用して、SOAPと呼ばれるXML形式の
プロトコルを用いメッセージの送受信を行う技術、またはそれを適用したサービス。[Wikipediaより]

Responder

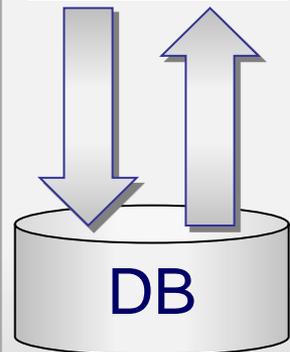
Requester



Web Service
Engine

Web サーバ

※WebサーバはTomcatが
iDempiereにはバンドルさ
れている。



DB



レスポンス

リクエスト

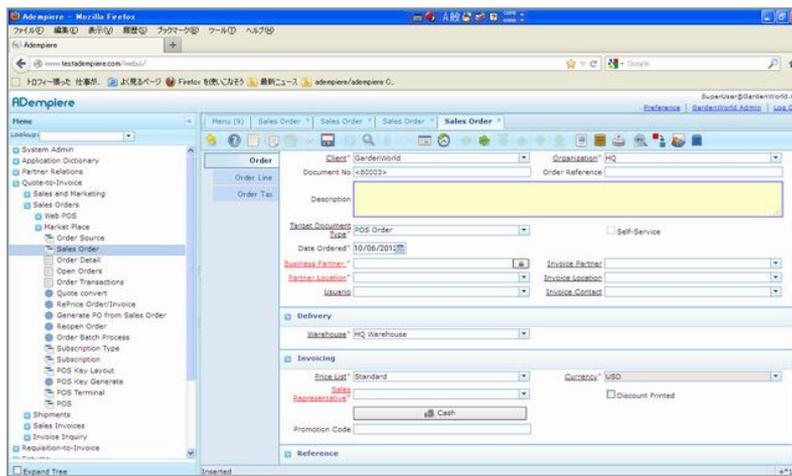
特徴的な機能:GUIその① (2種類あるGUI)



Swing-UI クライアントサーバー型のGUIで、JavaのSwingコンポーネントで作られている。



Web-UI OSS-ERPをWebアプリケーションとして使えるように作られたUI。



特徴的な機能:GUIその② (表示の切り替えとタブ構成)



単票表示とグリッド表示の切り替え



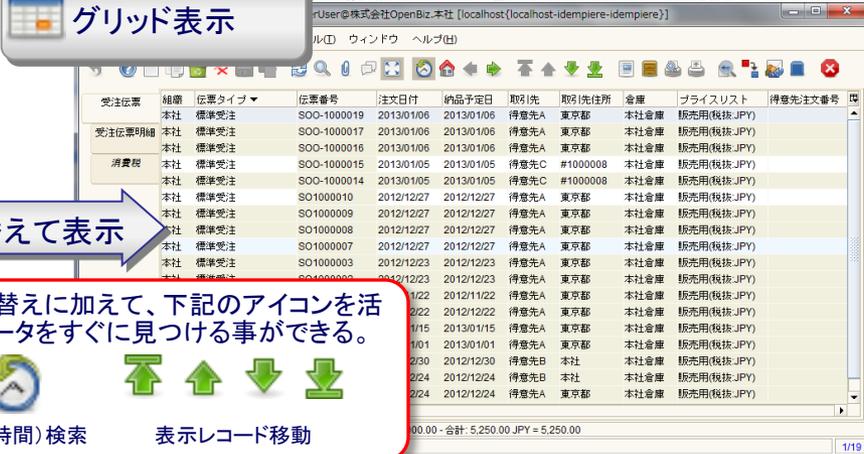
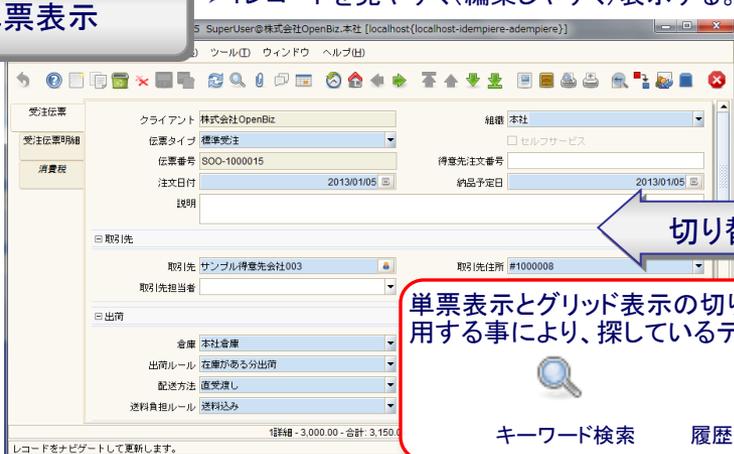
単票表示

>1レコードを見やすく(編集しやすく)表示する。



グリッド表示

>複数レコードをエクセルのように表示する。



切り替えて表示

単票表示とグリッド表示の切り替えに加えて、下記のアイコンを活用する事により、探しているデータを見つかる事ができる。



キーワード検索



履歴(時間)検索



表示レコード移動



タブ(Tab)形式のGUI



標準画面は複数のタブにより構成されており、タブ間には親子関係がある。

(例)

受注伝票(ヘッダ情報) **親**

受注伝票明細(明細情報) **子**

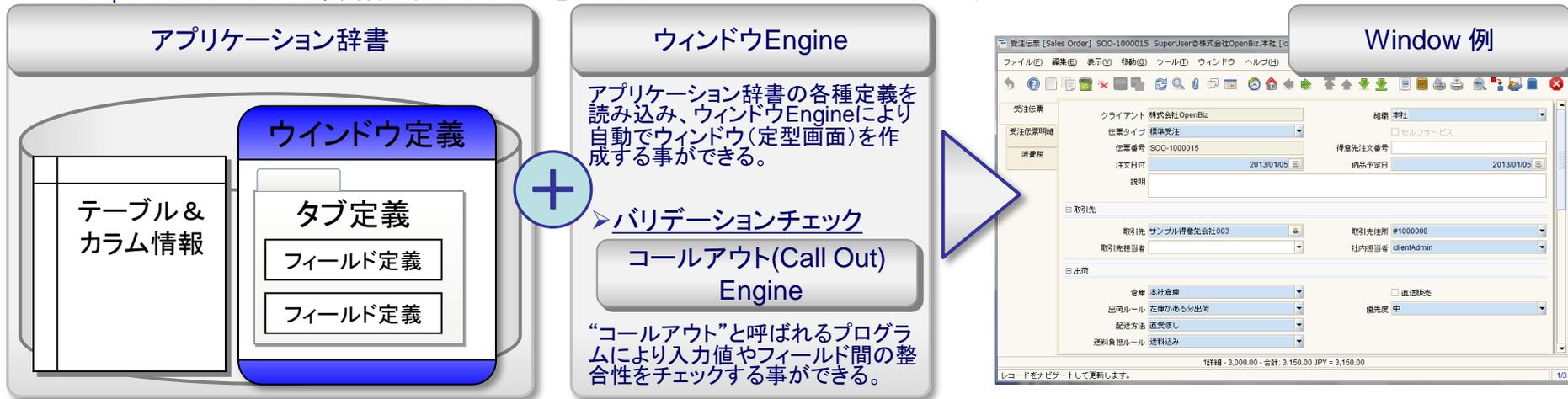
消費税(消費税情報) **子**

※1つのタブが1つのテーブルに対応している。

開発手法:GUI (ウィンドウとフォーム)

ウィンドウ(Window) ⇒ 定型画面(標準画面)

- ウィンドウ(Window)はADempiereが自動作成する定型の画面で、アプリケーション辞書に格納されているテーブルとカラムの情報を基礎とし、ウィンドウ定義でウィンドウを構成するタブの数や表示項目、表示順序などを設定する事により、ウィンドウエンジンが自動作成する。
- ADempiereではデータの登録/更新/削除はウィンドウを作成するだけでノンプログラミングで行える。



フォーム(Form) ⇒ 非定型画面

定型画面(ウィンドウ)ではなく自由に画面を作成したい場合に、ADempiereでは、フォームという概念が用意されており、作成した画面を簡単に組み込む事ができるようになっている。

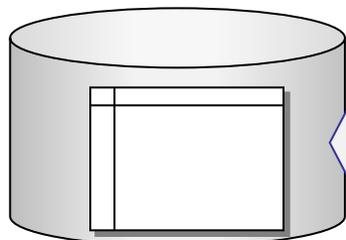


開発手法:ビジネスロジックの実装



データの登録・更新・削除時のビジネスロジックの実装

- ADempiereでは、データを登録・更新・削除した時に独自のビジネスロジックを実行する事ができる。
- 伝票(Document)として使用する場合には、伝票ステータス更新プロセス毎にビジネスロジックを実行する事ができる。

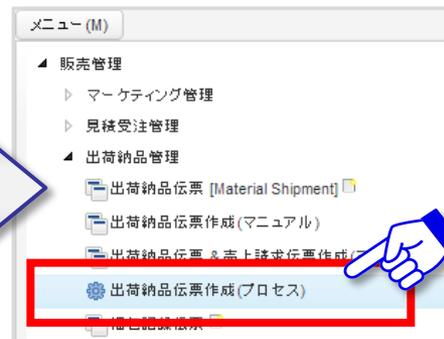
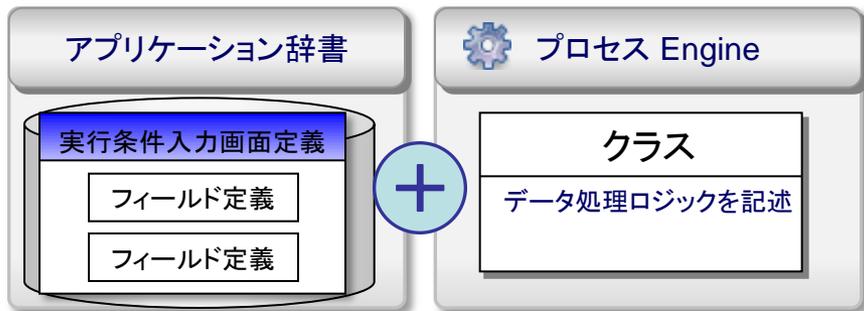


1つのテーブルに対して、1組のモデルクラスが作成され、そのテーブルに対するデータ処理(登録、更新、削除)の責任を負う。



プロセス(バッチ処理)としてのビジネスロジックの実装

- ADempiereではバッチ処理の事をプロセスとよび、メニューツリー上歯車のアイコンで表示される。
- ADempiereでは自作のバッチ処理をプロセスとして簡単にメニューに組み込む事ができる。



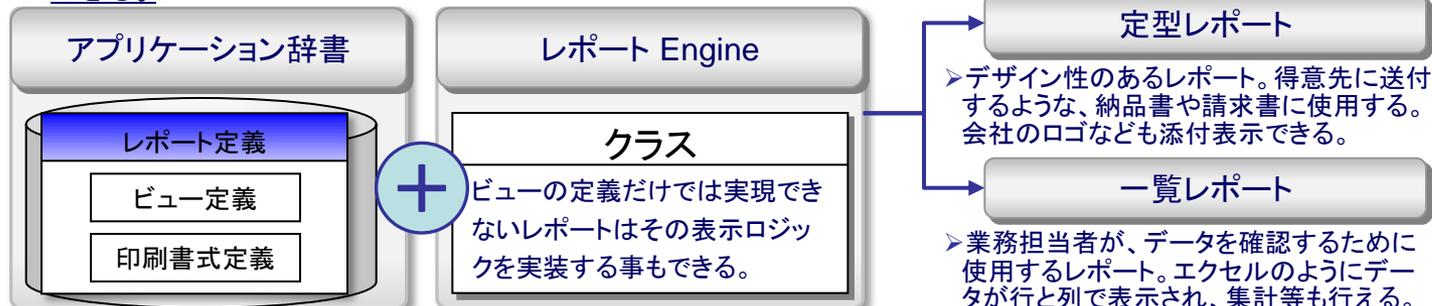
プロセス実行条件 入力画面例



開発手法: レポート

レポートEngine [開発者向けレポート作成ツール] を使用したレポート作成

管理&分析のためにデータを一覧表示する“一覧レポート”と予めデザインされた“定型レポート”をノンプログラミングで作成することができる。



(例) 定型レポート

請求書

行番	品名	数量	単価	金額	税別
10	商品A	10	2000.00	20000.00	
20	SAMPLE請求001	5	3000.00	15000.00	
30	SAMPLE請求002	2	5000.00	10000.00	
9999					
9999	トータル(9%)			45000.00	2250.00

集計	
小計	45,000.00円
消費税	2,250.00円
合計	47,250.00円

請求書 (例) 一覧レポート ページ 1(1,1) of 6(1,6)

パラメータ:

伝票No	注文書	注文日付	顧客注文番号	明
100000	80000 - 2003-11-01 00.00.00	2003/11/01		Export Order
200000	80000 - 2002-02-22 00.00.00	2002/02/22		(1)
200001	80001 - 2002-02-22 00.00.00	2002/02/22		(2)
200002	80002 - 2002-08-01 00.00.00	2002/08/01	Phone Order	Text is printed in the language of the business partner (if not entered in English like this). This is also a text, which will be printed in block format on two lines on the documents.

財務レポート (Financial Report) [パワーユーザー向けレポート作成ツール]でのレポート作成

- 仕訳のデータをもとに財務部・経理部のパワーユーザーがレポートを作成することができる簡易レポート作成ツール。
- 財務諸表としてBS、PL、CFの作成のほか、管理会計用の多次元分析レポートが作成できるレポート作成ツール。

Jasper Reportを活用したレポート作成

開発手法:業務アプリケーションの開発基盤として

- ▶ ADempiereは業務アプリケーションの開発基盤として、コーディングはビジネスロジック部分に集中して行えば良いシステムアーキテクチャになっている。GUIの作成やデータの登録/変更/削除のDB更新処理などビジネスロジックとは直接的には関係のない処理はADempiereの各種エンジンプログラムがアプリケーション辞書の情報をもとに処理してくれるようになっている。



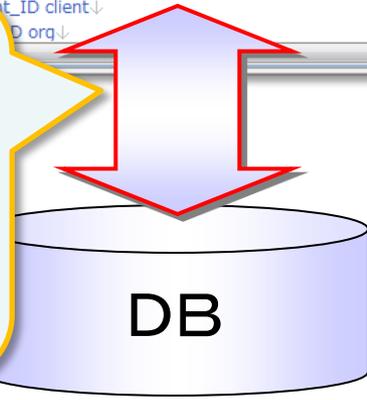
```
プロジェクト(P) Tomcat(T) 実行(R) Limy(L) ウィンドウ(W) ヘルプ(H)
Dimension2DImpl.java MOrder.java
org.compiere.model MOrder MOrder(Properties, ResultSet, String)
244 > /**↓
245 > * Load Constructor↓
246 > * @param ctx context↓
247 > * @param rs result set record↓
248 > * @param trxName transaction↓
249 > */↓
250 > public MOrder (Properties ctx, ResultSet rs, String trxName)↓
251 > {↓
252 > * super(ctx, rs, trxName);↓
253 > } // MOrder↓
254 ↓
255 > /** > Order Lines >>>> */↓
256 > private MOrderLine[] m_lines = null;↓
257 > /** > Tax Lines >>>> */↓
258 > private MOrderTax[] m_taxes = null;↓
259 > /** Force Creation of order > */↓
260 > private boolean >>> m_forceCreation = false;↓
261 ↓
262 > /**↓
263 > * Overwrite Client/Org if required↓
264 > * client_ID client↓
265 > * D org↓
```

ビジネスロジック
を中心に開発

Windowは自動作成され、GUIとのデータのやり取りはWindow Engineが処理し、DBへのデータの登録/更新/削除はPO(Persistent Object)クラスが処理してくれる。

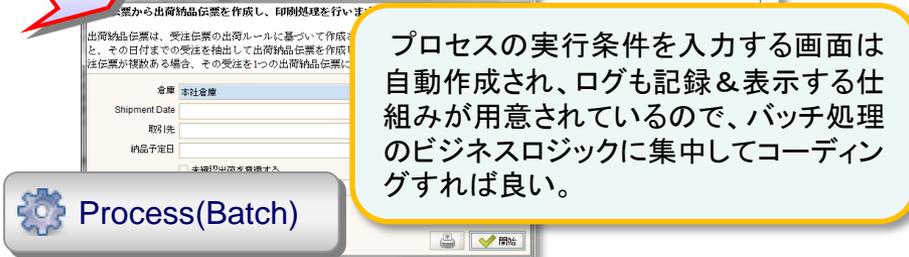
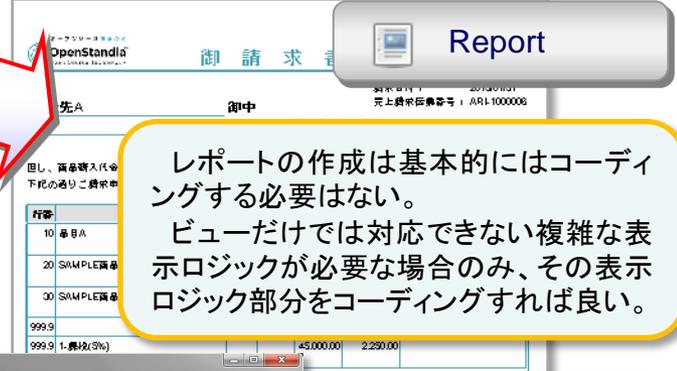
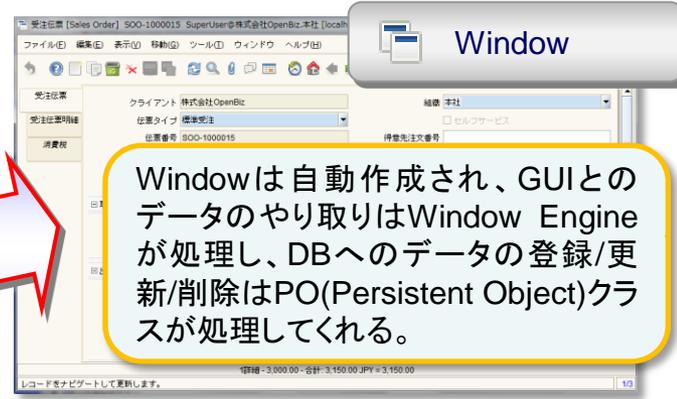
レポートの作成は基本的にはコーディングする必要はない。ビューだけでは対応できない複雑な表示ロジックが必要な場合のみ、その表示ロジック部分をコーディングすれば良い。

DBへのデータの登録/更新/削除の処理は、PO(Persistent Object)クラスが担うようになっている。ひとつのテーブルには対応するひと組のモデルクラスが作成され、そのモデルクラスはPOクラスを継承して作成される。開発者はビジネスロジックの中でモデルクラスのインスタンスを呼び出して必要に応じてsave()メソッドや、delete()メソッドを記述するだけで、DBへのデータの登録/更新/削除が行える。



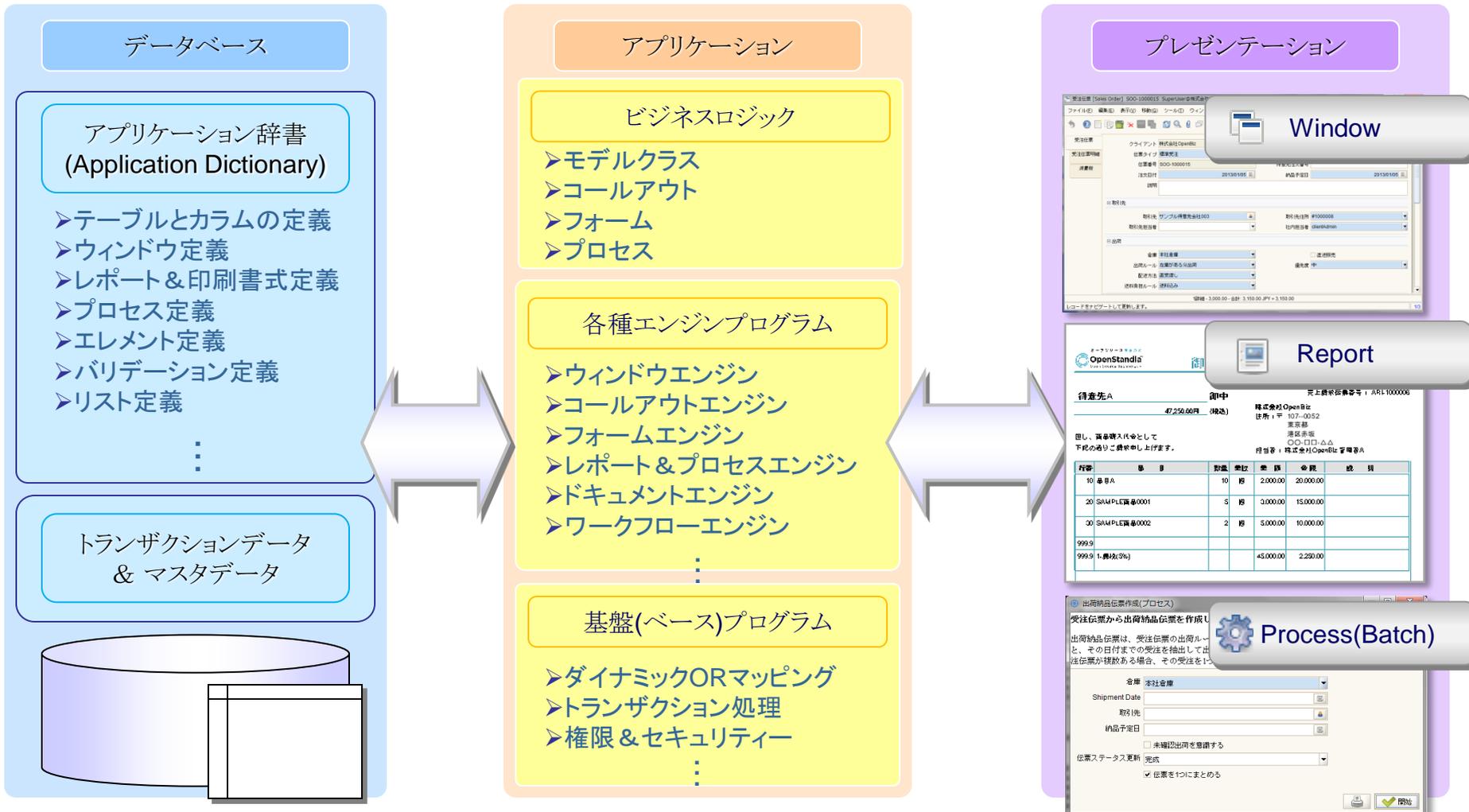
Process(Batch)

プロセスの実行条件を入力する画面は自動作成され、ログも記録&表示する仕組みが用意されているので、バッチ処理のビジネスロジックに集中してコーディングすれば良い。



開発手法: システムアーキテクチャイメージ

モデル駆動アーキテクチャ [Model-Driven Architecture(MDA)]



ADempiere活用のポイント(1)

“オープンソースの業務アプリケーション”という、新しい分野のアプリケーションを上手に活用することが、企業の競争力を高める1つの方法である。オープンソースの業務アプリケーションを活用できれば、他社と比べて大きなアドバンテージになるはずである!!



バージョンアップは気にせず自由にカスタマイズ

- バージョンアップを意識してシステム導入を行うと、通常はカスタマイズでModification(標準機能のソースコードを修正する事)が行えない(修正する事自体は可能だが、バージョンアップ時に上書きされて元に戻ってしまう可能性が高い)。Modificationは行わず、ADD-ONプログラムとパラメータ設定だけでカスタマイズを行うことは、オープンソースという特徴を活かしていると言えるのか？
- 会社毎にカスタマイズが必ずといって良いほど必要となる業務アプリケーションにおいて、オープンソースのメリットを最大限に享受する方法は、バージョンアップの事は気にせず、導入時に導入企業の要件に合うようにModificationも含めたカスタマイズを自由に施し、その後はビジネスの変化に応じて必要な機能が出てきたら、追加開発してシステムを運用していく方法なのではないか!?
- この考え方は、オープンソースの業務アプリケーションをスクラッチ開発の基盤(業務アプリケーションの開発基盤)と考えると理解しやすい。スクラッチ開発するシステムにおいて、バージョンアップされたシステムがどこからか提供されるということは無い。スクラッチ開発したシステムは必要な時に機能を追加開発し運用していくのが通常だろう。オープンソースの業務アプリケーションをスクラッチ開発の基盤だと考えれば、開発基盤なしにフルスクラッチ開発したシステムよりも、低コスト/短納期/低リスクで高機能な業務アプリケーションを作る事ができるのでオープンソースの業務アプリケーションのメリットを享受できていると言えるのではないか!!
- 将来的にどのような機能が追加されるかわからないバージョンアップを期待してオープンソースと言えどもライセンス料を支払ったり、バージョンアップを考慮して、Modificationで簡単に実現できる機能をADD-ONプログラムで開発するのは、正しい判断なのか？バージョンアップを意識してModificationは行わず、その代わりにADD-ONとして開発された機能は皮肉にもバージョンアップ時には動作確認などの工数発生要因にもなってしまう。
- 商用ERPを導入している企業の多くはバージョンアップをサポート期間切れなどのネガティブな理由でしかたなく行っているのが実情である。商用ERPを使用している企業でさえ本音ではバージョンアップはできれば避けたいと思っているのではないか？そうであれば、なおさらオープンソースの業務アプリケーションはバージョンアップは気にせず自由にカスタマイズを施して運用して行った方が良いのではないか。バージョンアップに関わるコストの削減にもつながる。
- バージョンアップは気にせず10年・20年と使い続けて、バージョンアップのメリットがデメリットを上回った時に、その時の最新のバージョンで”再構築(データ移行&追加開発した機能の移植 +α)”すれば良いくらいの心持ちで良いのではないか。導入企業は使い慣れたシステムの最新版で業務システムを再構築するので、社内に蓄積されている活用ノウハウが活かせ、より良いシステムのバージョンアップになるはずである。

ADempiere活用のポイント(2)



単純に機能面だけで商用業務アプリケーションとOSSの業務アプリケーションを比較しない。

- 機能面で商用ERPとOSS-ERPを比較する質問を受ける事があるが、既に実装されている標準機能だけで単純に比較するのであれば有名商用ERPの方が色々な機能が備わっている事だろう。ただ、有名商用ERPの場合には使用しない機能に対しても多くの金額を支払っていると考える事ができる。OSSの業務アプリケーションは自由に追加開発できる事が大きなメリット(特長)であり、足りない機能は自分達で開発する事が前提とも言える。特にADempiereの場合は、追加開発が非常にしやすくなっている。商用業務アプリケーションとOSSの業務アプリケーションを単純に1つの機能のある／なしで比較するのではなく、比較するのであれば、必要な機能は追加開発するという前提で、TCO(*Total Cost of Ownership / Ownership*)の観点から比較すべきである。



使える所(使える機能)を使う。

- ADempiereはERPとして非常に多くの機能を有しているが、無理にERPとして全ての機能を使用する必要はない。OSSで無料という特性を生かして販売管理システムとして導入したり(請求書を発行し、債権の管理のみ行うのもOK)、会計システムとして導入したり、自社で利用できる部分を利用するという方針でも良いと思う。



困ったときに頼れるサポート体制を構築しておく。

- オープンソースは自己責任とよくセットで語られるが、オープンソースのサポートを提供している会社はいくつも存在している。困った時に頼れるサポート体制を予め構築しておくのも、業務システムとしては必要な事である。

ADempiere活用のポイント(3)

 他のオープンソースの業務アプリケーションとの連携を考える。

- ADempiereだけではなく、他のOSS業務アプリケーションと連携する事で、コストを最小限に抑えたIT経営の推進が可能!!

