

本格導入が始まったオープンソースERP(ADempiere/iDempiere) オープンソースERPをフル活用するための

10のポイント



野村総合研究所 情報技術本部
オープンソースソリューション推進室



野村総合研究所のOpenStandia（オープンスタンディア）は、おかげさまで、2006年のサービス開始から2011年までの5年間で契約数累計が1,000件を突破いたしました！

株式会社 野村総合研究所 情報技術本部 オープンソースソリューション推進室

Mail : ossc@nri.co.jp Web: <http://openstandia.jp/>



オープンソースERPのご紹介

- ADempiere(アデンピエレ)/iDempiere(アイデンピエレ)とは
- 業務機能概要

 【ポイント1】 OSS業務アプリならではの違いを理解する

 【ポイント2】 コスト削減のために使える機能を使用する

 【ポイント3】 TCOで考える

 【ポイント4】 他のOSSと組み合わせてIT経営を促進する

 【ポイント5】 グループ経営管理に活用する

 【ポイント6】 業務機能を理解した上で開発手法を学ぶ

 【ポイント7】 自由にカスタマイズする

 【ポイント8】 導入に適している条件を知る

 【ポイント9】 OSS業務アプリのメリットとデメリットを理解する

 【ポイント10】 困った時に頼れるサポート体制を構築する

【機能紹介】

- ・内部統制
- ・I/F
- ・システムの枠組み
- ・グローバル対応
- ・GUI など

ADempiere(アデンピエレ)/iDempiere(アイデンピエレ)とは

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

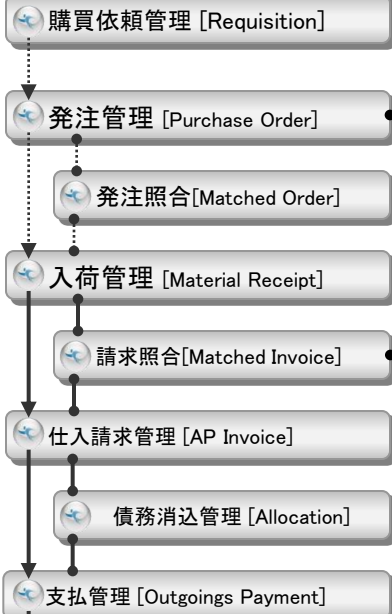
-  購買管理、在庫管理、販売管理、生産管理、会計管理、顧客管理など、一連の業務機能が既に実装されている、無料で使用できるオープンソースのERP。
-  中小企業・大企業の子会社などを主なターゲットとした、無料で使用できるオープンソースのERP。
-  1企業で導入できるのはもちろん、グループ企業(複数企業)にも導入できる、無料で使用できるオープンソースのERP。
-  世界中で使用されておりグローバル対応されている、無料で使用できるオープンソースのERP。
-  クラウド環境でも利用できる、無料で使用できるオープンソースのERP。
-  Compiere ⇒ ADempiere ⇒ iDempiere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

購買管理



在庫管理



生産管理



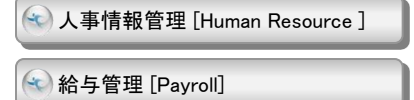
販売管理



顧客管理(CRM)



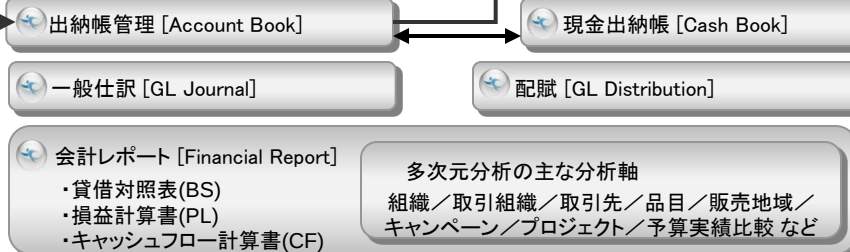
人事給与管理



その他の業務機能



固定資産会計



会計管理(財務会計 & 管理会計)

【ポイント1】 OSS業務アプリならではの違いを理解する



「OSSのOSやミドルウェア(DB等)」と「OSSの業務アプリケーション」の違いを理解しよう!!

Q1. 「OSSのOSやミドルウェア(DB等)」と「OSSの業務アプリケーション」の大きな違いは何か。

A.導入企業の要件に応じてカスタマイズが必要になるかどうか。

- 同じオープンソースでもLinuxなどのOSやPostgreSQLなどのミドルウェアは、導入企業の要件に応じてチューニング作業などはあっても、ソースコードを追加/修正するような開発を伴うカスタマイズを行う事は通常は無い。しかし、業務アプリケーションは導入企業の要件に応じてカスタマイズが必要になってくる。ソースコードを自由に追加/修正して好きなようにカスタマイズができるというオープンソース本来のメリットは業務アプリケーションにこそ発揮される。

カスタマイズの種類

Add-On (アドオン)

既存のソースコードは修正せずに、機能を追加開発する事。

Modification (モディフィケーション)

既存のソースコードを直接修正する事。

Parameter / Configuration (パラメータ設定／コンフィグ設定)

パラメータの設定で、カスタマイズする事。

【ポイント1】 OSS業務アプリならではの違いを理解する

「有料の商用業務アプリ」と「OSSの業務アプリ」の導入手法の違いについて考えよう!!

Q2. 「商用業務アプリ」と「OSSの業務アプリ」で導入に対する考え方(SIの方法論)や取り組みは同じで良いのか？

- 商用業務アプリケーションをカスタマイズする場合、パラメータ設定／コンフィグ設定が主で、ソースコードを直接修整するようなModificationは通常はサポート対象外になってしまうため行わない。Add-On開発も極力少なくしろと言われる。オープンソースの業務アプリケーションも同じような考えで導入プロジェクトを推進すべきか？
- 商用業務アプリケーションでは、業務をアプリケーションに合わせるようにコンサルティングされるが、オープンソースの業務アプリケーションも同様の考え方で良いのか？
- ウォーターフォール型の開発モデルが最適なのか？

カスタマイズの種類

Add-On (アドオン)

既存のソースコードは修正せずに、機能を追加開発する事。

Modification (モディフィケーション)

既存のソースコードを直接修正する事。

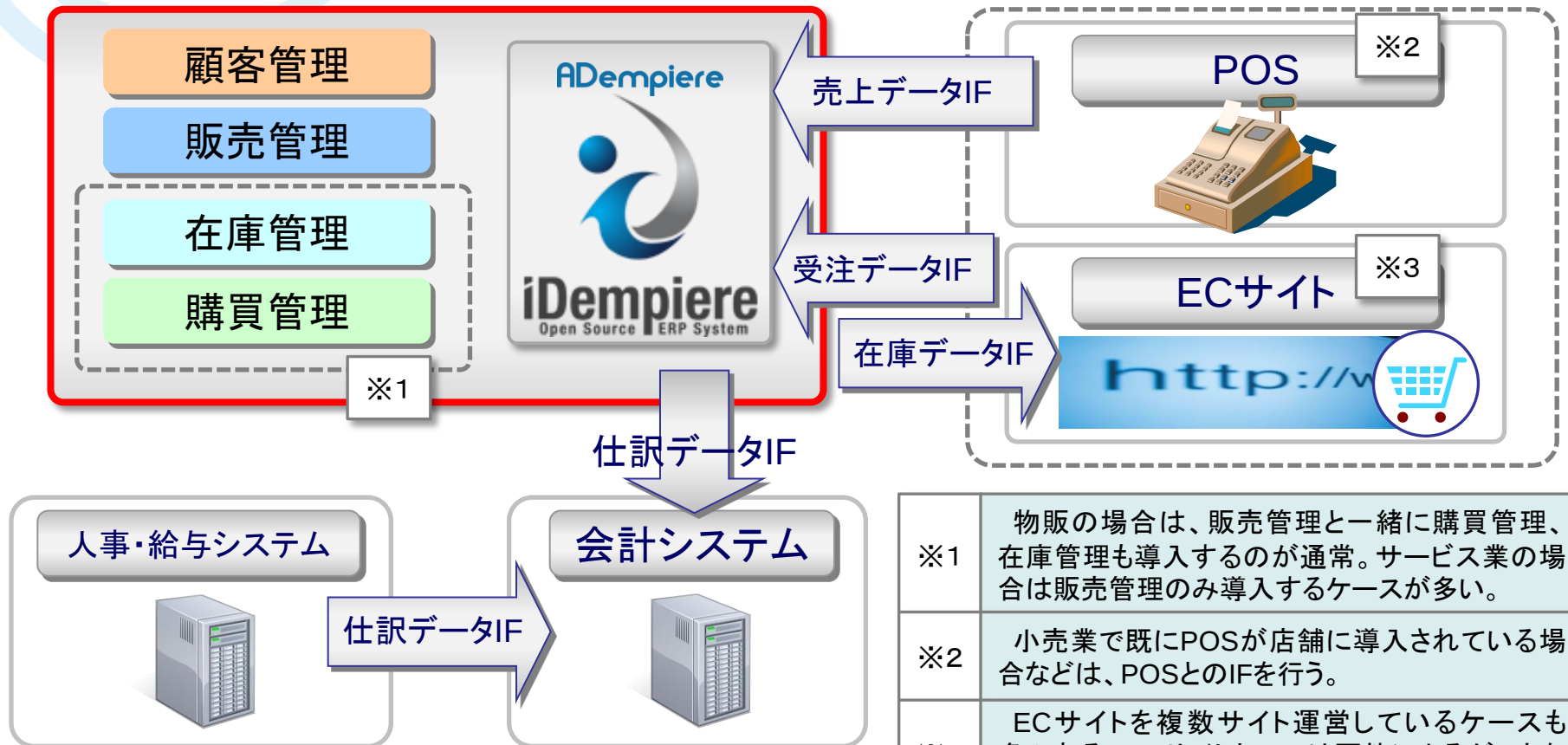
Parameter / Configuration (パラメータ設定／コンフィグ設定)

パラメータの設定で、カスタマイズする事。

【ポイント2】 コスト削減のため使える機能を使用する

- ADempiere/iDempiereはERPであるが、無理にERPとして使用する必要は無い。オープンソースで“無料”という特長を活かして、販売管理や在庫管理システム、会計システムなどとして自社で活用できる部分を利用するという方針でも良いのではないか。

ADempiere/iDempiereを販売管理システムとして使用するケース



※1	物販の場合は、販売管理と一緒に購買管理、在庫管理も導入するのが通常。サービス業の場合は販売管理のみ導入するケースが多い。
※2	小売業で既にPOSが店舗に導入されている場合などは、POSとのIFを行う。
※3	ECサイトを複数サイト運営しているケースも多々ある。ECサイトとのIFは要件によるが、少なくとも受注データのIFは必要。

* IF(I/F)→interface (インターフェース) の略

【ポイント2】コスト削減のため使える機能を使用する

- ADempiere/iDempiereを使用するのに決まったルールや制限はない。好きなように使用できるのがOSSの良いところなので、既成概念にとらわれず、自由に活用して行きたい!!

リクエスト管理を活用した 小規模コールセンター

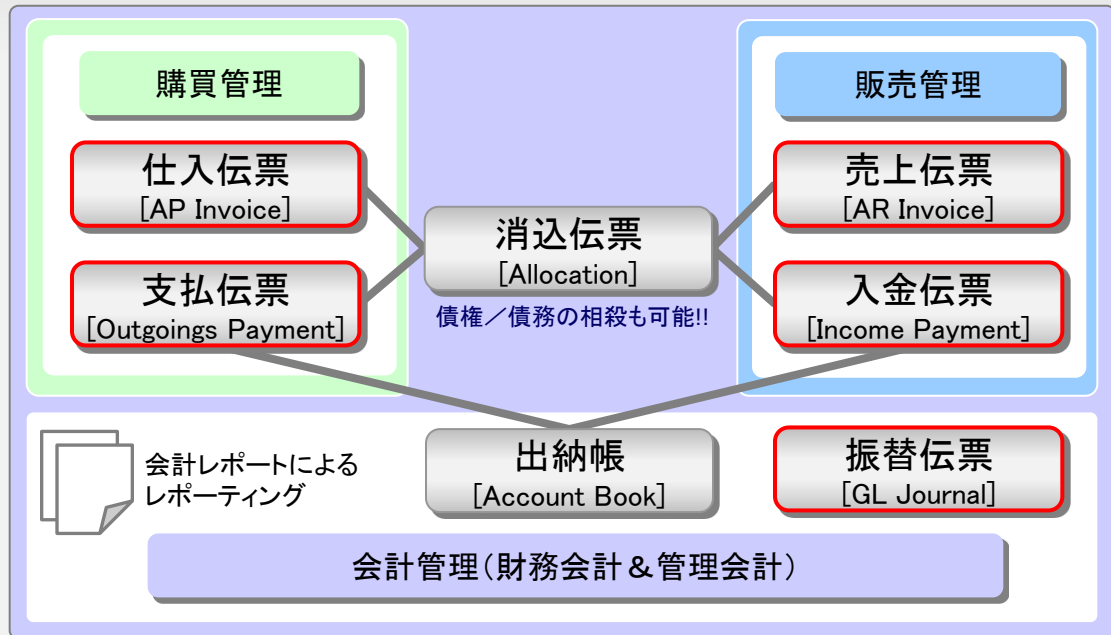


リクエスト管理

顧客の要求(リクエスト/クレーム)をADempiere/iDempiereのマスタやトランザクションデータと結びつけて管理し、社内でも共有する事ができる機能。

ADempiere/iDempiereのCRMの中核機能の1つで、顧客からの問い合わせ履歴管理など小規模なコールセンターに適用できる機能が揃っている。

(五伝票制の伝票会計的な) 会計システム



会計モジュールの機能だけでも会計システムとして運用していく事ができるが、販売管理の売上(請求)伝票や入金伝票、購買管理の仕入(請求)伝票や支払伝票(出金伝票)等を使用する事により五伝票制の伝票会計的な会計システムの運用が可能になる。会計モジュールだけの運用より、債権/債務の消込管理などが行え、管理レベルが向上する。

【ポイント3】 TCOで考える

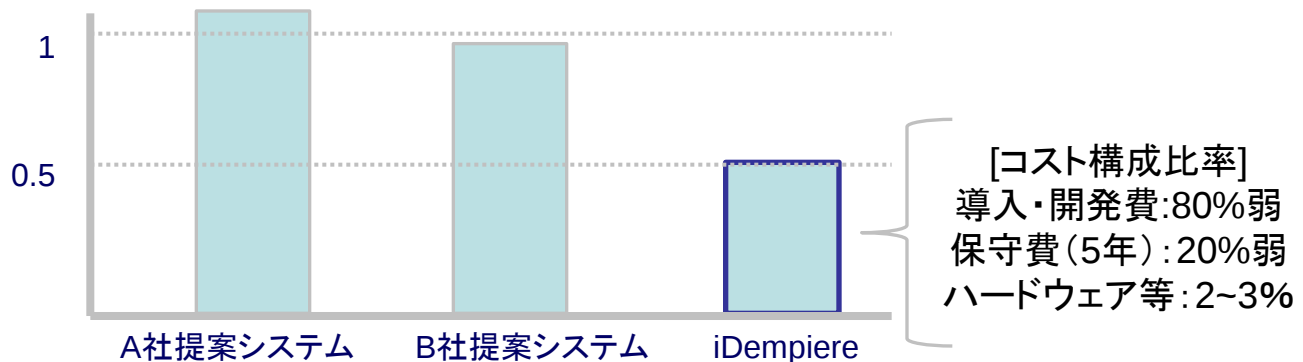
単純に既に実装されている標準機能で、有名商用ERPとOSSのERP (ADempiere/iDempiere)を比較するのであれば、有名商用ERPの方が色々な機能が備わっている事だろう。ただ、有名商用ERPの場合は使用していない機能に対しても多くの費用を支払っていると考える事ができる。OSSのERPは、自由にカスタマイズできる事が大きなメリット(特長)であり、足りない機能は自分達で開発する事が前提とも言える。特にADempiere/iDempiereの場合は、カスタマイズが非常にしやすくなっているので、比較するのであれば、必要な機能は追加開発するという前提で、**TCO(Total Cost Operation/Ownership)**の観点で比較検討すべきである。



ある企業のiDempiere選定理由(販売管理システムとして導入)

● 費用対効果(5年間のトータルコストが約半分!!)

はじめは5~6社に提案を求め、1次選考で3社に絞り込む。その3社に対して、導入費用や保守、ハード費用も含めて5年間のトータルコストの見積を取得。その結果、他の2社の約半分のコストであった。



● カスタマイズの自由度と柔軟性

オープンソースは標準機能においてもソースコードの修正が可能であり、標準機能と業務で要求される機能との間にギャップがあった場合でも、柔軟に適合させる事が可能。

● 内部統制の強化

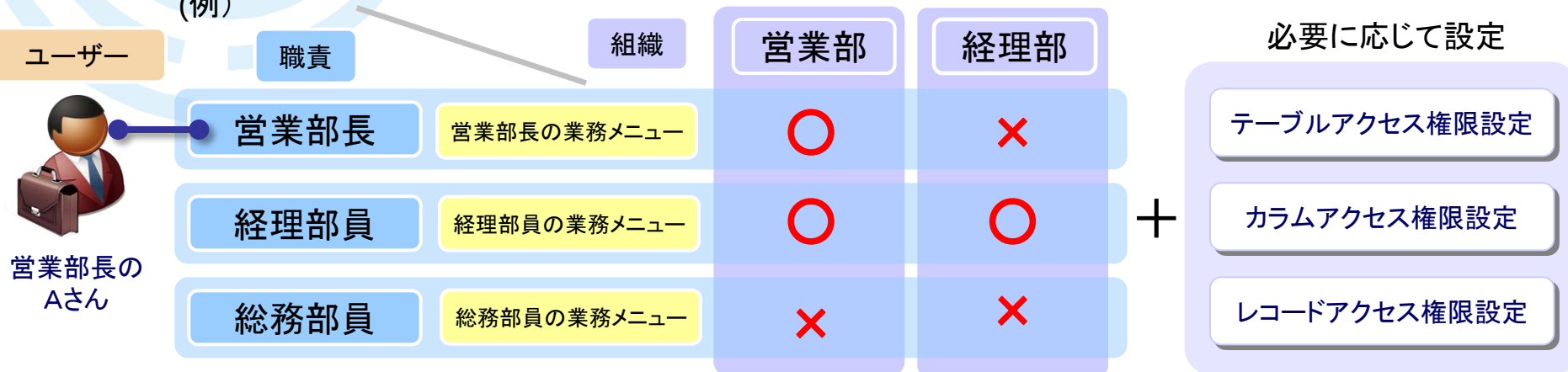
株式公開を目指しており、これまでのシステムでは内部統制上の不安があった。

【機能紹介】 内部統制

権限管理

- ADempiere/iDempiere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

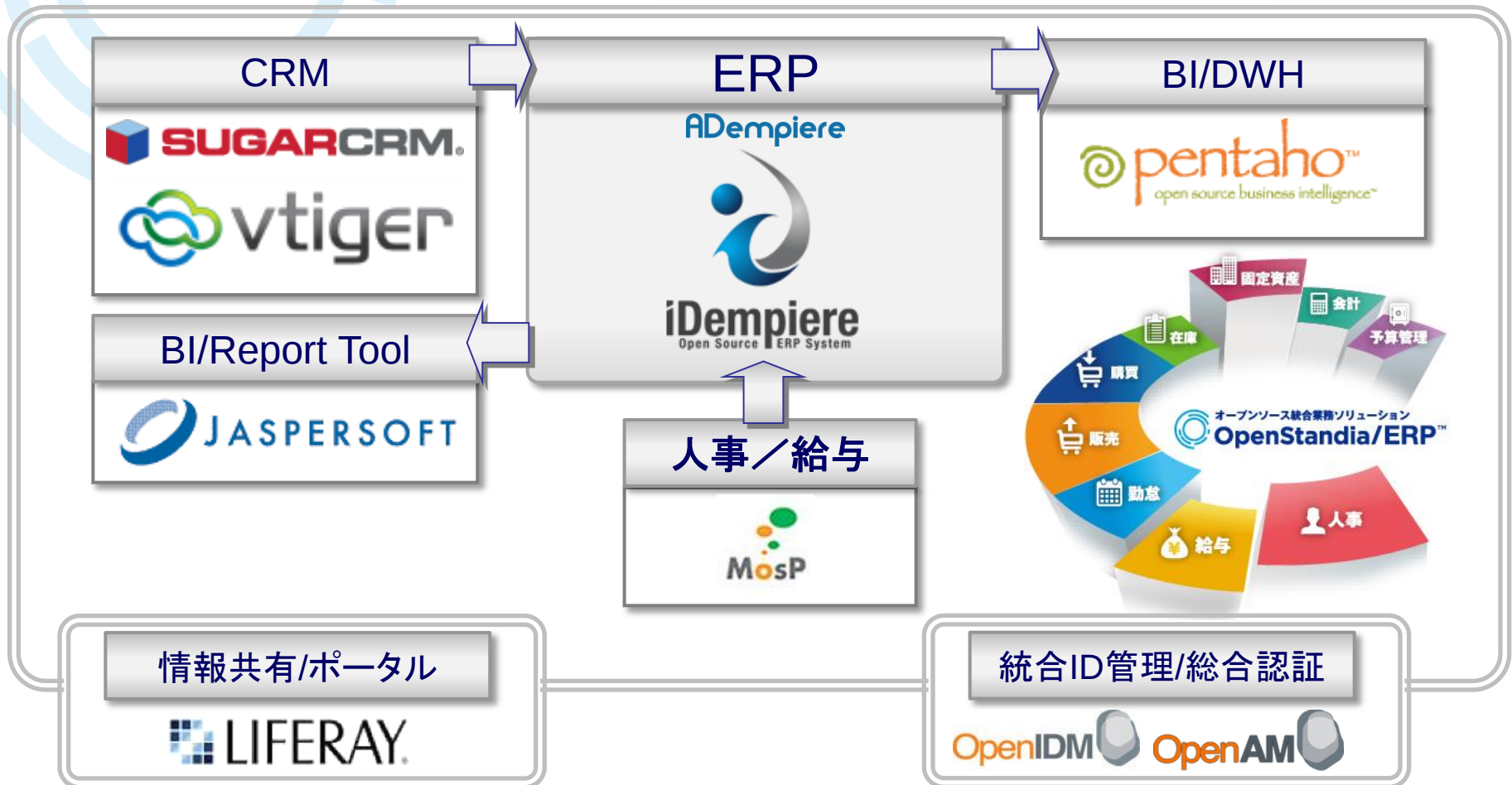
常にログ管理の対象

“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。



【ポイント4】 他のOSSと組み合わせてIT経営を推進

- ADempiere/iDempiere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!!





データのエクスポート

Postscript

XML

PDF

HTML

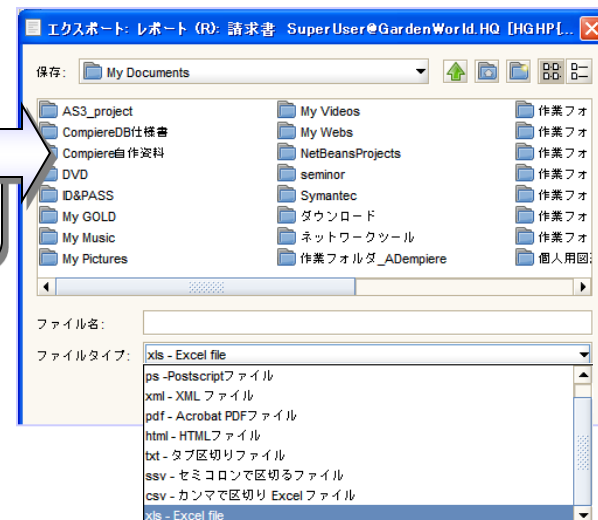
タブ区切り

セミコロン区切り

csvカンマ区切り

xls

エクスポート



データのインポート

インポート定義

テーブル

カラム

インポート
Engine

データ インポート

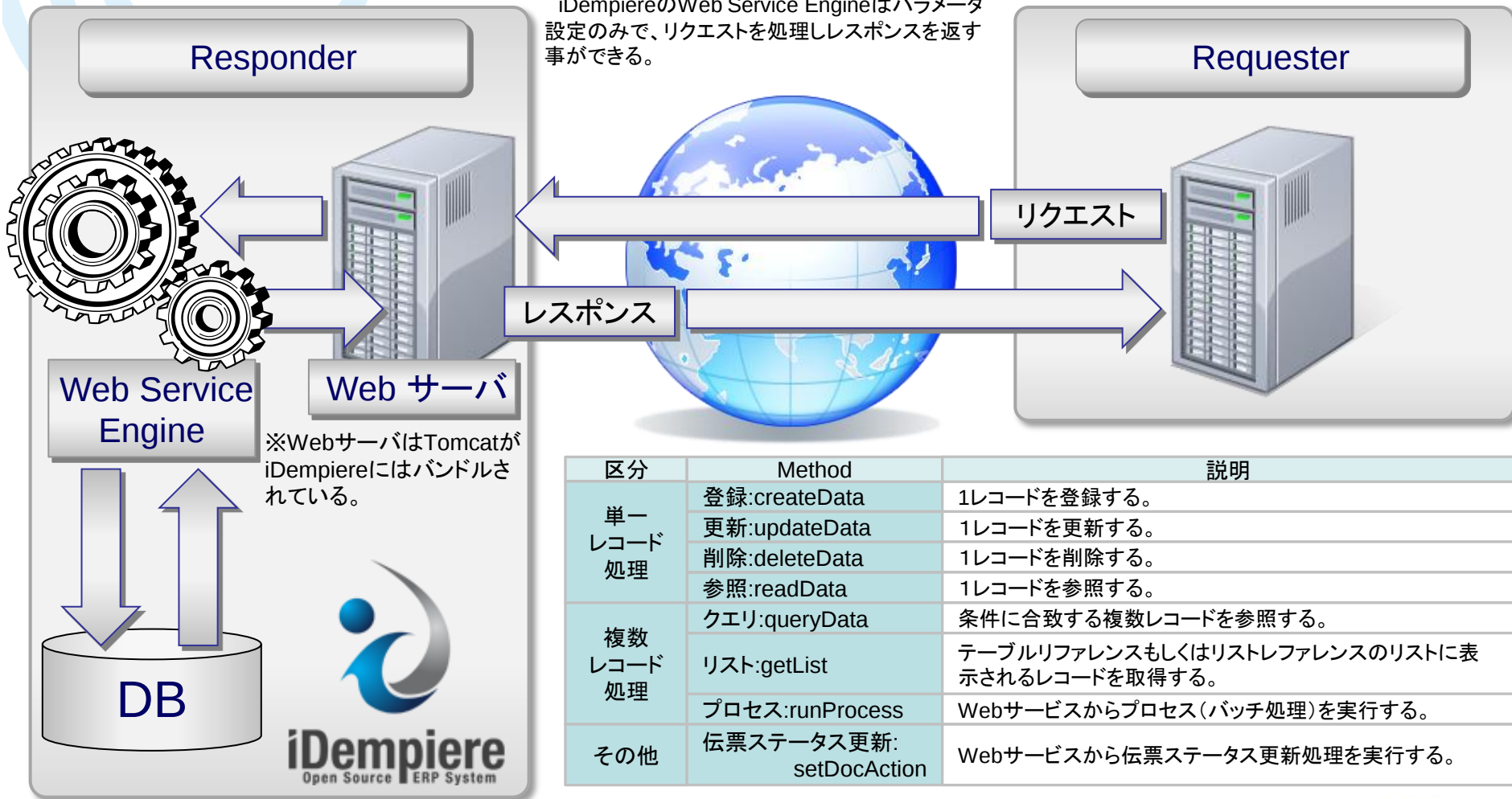
【機能紹介】 Web Service



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]

iDempiereのWeb Service Engineはパラメータ設定のみで、リクエストを処理しレスポンスを返す事ができる。



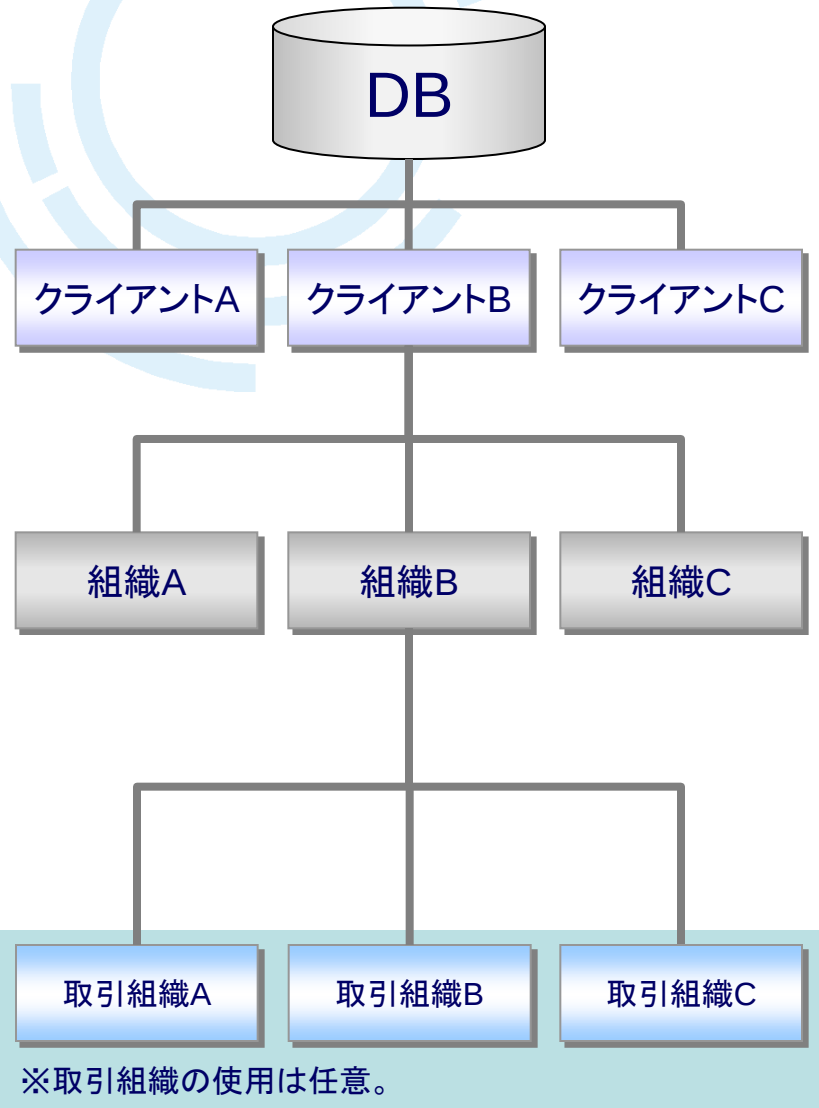
【ポイント5】 グループ経営管理に活用する

- 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設計されている。
- リアルタイムでグループ企業全体の状況把握が可能に!!

 グループ企業



【機能紹介】 システムの枠組み



ADempiere/iDempiereの枠組み

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セグメントデータを管理する事ができる。

【ポイント5】 グループ経営管理に活用する

[発展] フランチャイズ企業や販売会社の管理に活用する

➤ グループ企業だけでなく、フランチャイズ企業や販売会社の管理にも活用できる!!

- トランザクションデータは組織毎に区分管理されるので、資本関係に関わらず1つのシステム内で管理が可能!!
- グループ企業やフランチャイズ企業からシステム利用料を徴収し、コスト回収するケースも多い。
- 店舗展開しているような業態に!!

 グループ企業

 フランチャイズ企業
又は販売会社

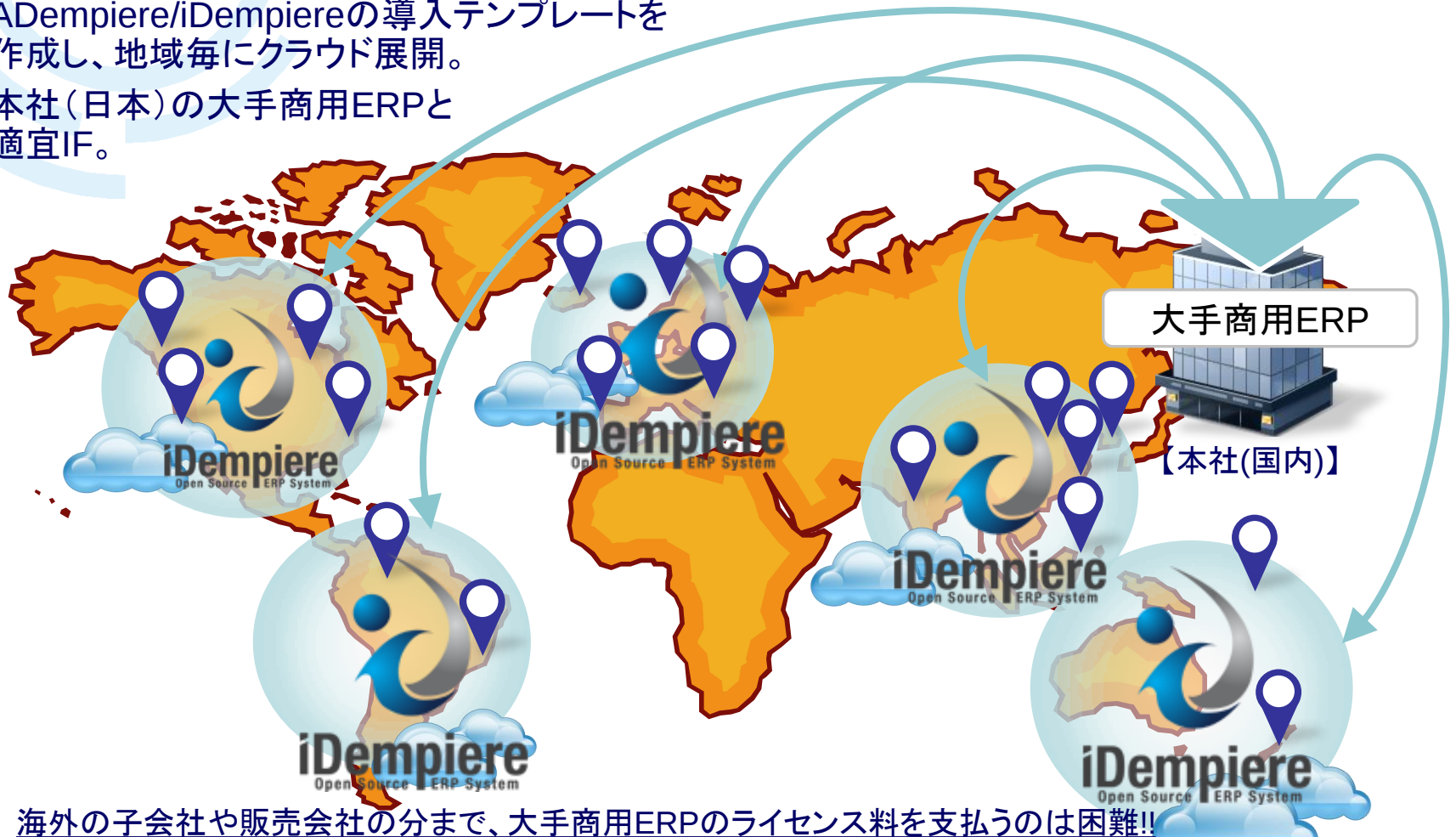


【本社】

【ポイント5】 グループ経営管理に活用する

[発展] グローバルに活用する

- ADempiere/iDempiereの導入テンプレートを作成し、地域毎にクラウド展開。
- 本社(日本)の大手商用ERPと適宜IF。



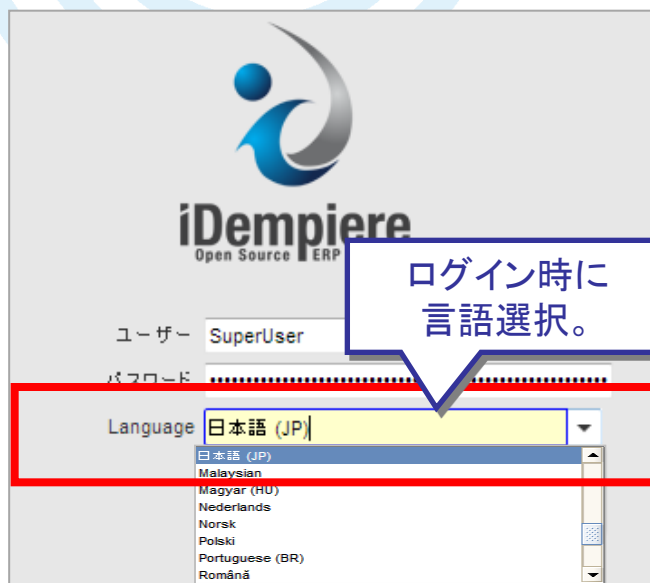
- 海外の子会社や販売会社の分まで、大手商用ERPのライセンス料を支払うのは困難!!
- 導入テンプレートの作成とクラウド対応で海外事業展開のスピードアップ!!

【機能紹介】 グローバル対応



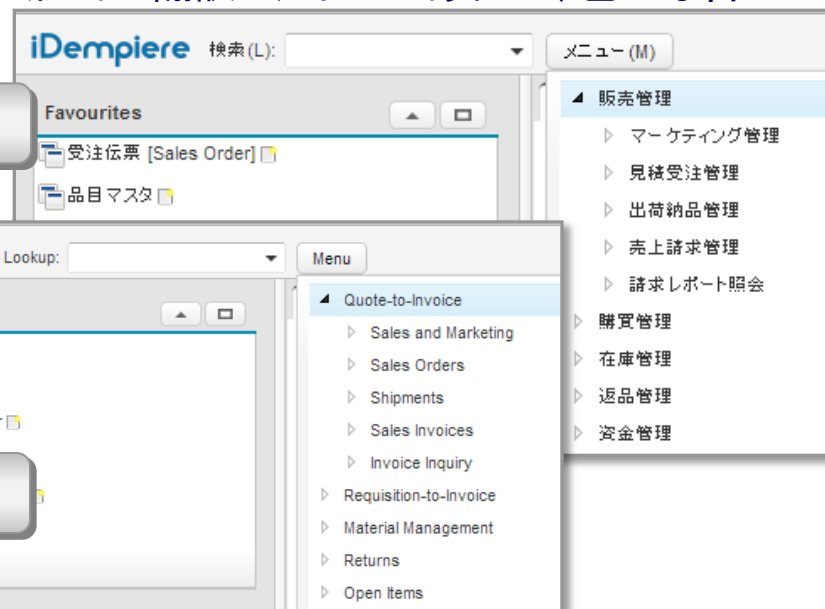
多言語対応(マルチ言語)

- 100を超える言語のマスタが既に設定されており、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。



日本語

英語



※翻訳ファイルが無くてもシステム上で翻訳作業を行っていく事もできる!!



多通貨対応(マルチ通貨)

- 170を超える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。

【機能紹介】 グローバル対応 & クラウド対応

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/iDempiere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
 - 地域別消費税設定
 - 消費税等の複合処理
 - “伝票明細単位での消費税計算”と“伝票単位での消費税計算”の選択適用
 - 税率改正対応
 - 営業税として費用処理
 - 税込価格と税抜価格の処理
 - 個別税抜方式の自動仕訳 など...

クラウド対応

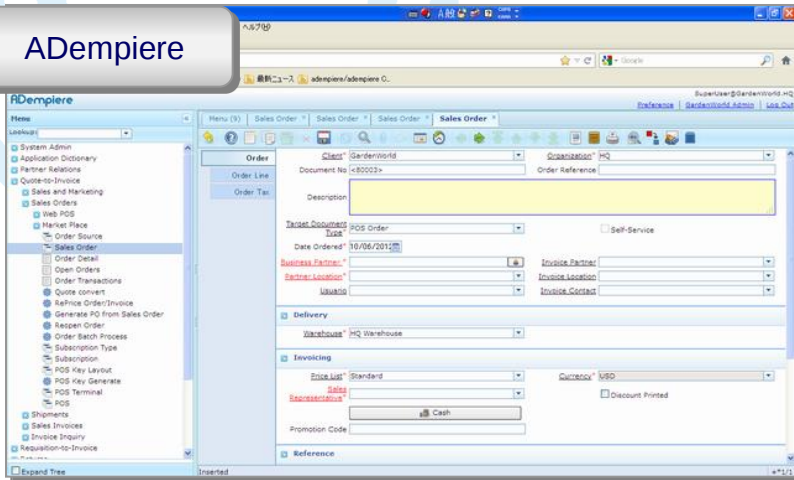
- ADempiere/iDempiereはクラウド環境でも利用できる(≒Webアプリケーションとして使用することができる)、無料で使用できるオープンソースのERP。

【機能紹介】GUIその①:2種類あるGUI

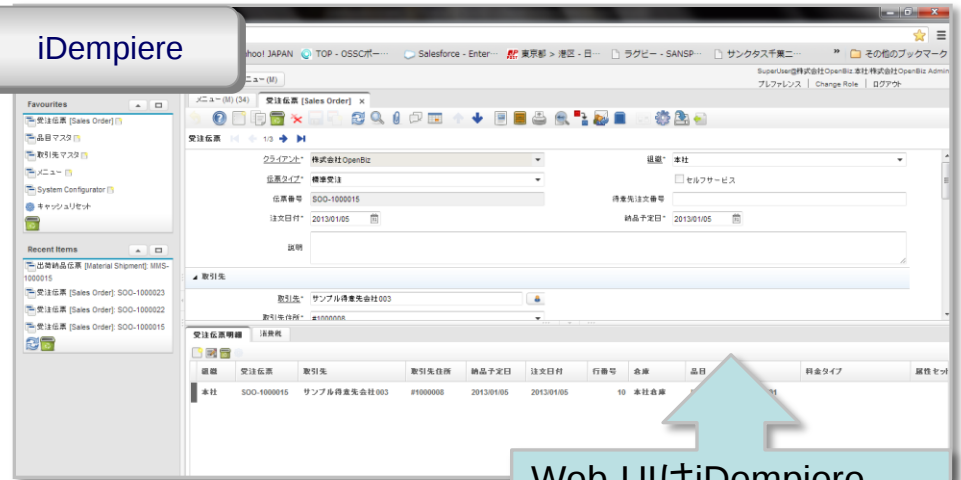


Web-UI ADempiere / iDempiereをWebアプリケーションとして使えるように作られたUI。

ADempiere



iDempiere



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はiDempiere
になって、大幅に機能が
強化されている。

➤1レコードを見やすく(編集しやすく)表示する。

➤ 複数レコードをエクセルのように表示する。

[illegible]

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すぐに見つける事ができる。



キーワード検索



履歷(時間)檢索



表示レコード移動

受注伝票 [Sales Order] SOO-1000015 SuperUser@株式会社OpenBiz.本社 [localhost[localhost~admpiere-admpiere]]

ファイル(F) 編集(E) 表示(V) 移動(M) ツール(T) ウィンドウ W ヘルプ(H)

受注伝票

受注伝票明細

消費税

クライアント 株式会社OpenBiz

依頼タイプ 標準受注

伝票番号 SOO-1000015

注文日付 2013/01/05

説明

組織 本社

得意先 待意先注文番号

納品予定日 2013/01/05

取引先 サンプル得意先003

取引先住所 #1000008

倉庫 本社倉庫

出荷ルール 在庫がある分出荷

配送方法 直受渡し

送料負担ルール 送料込み

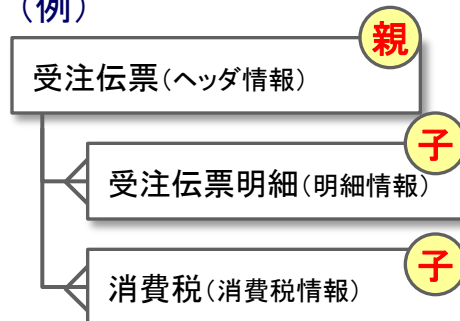
優先度 中

見積額 - 3,000.00 - 合計: 3,150.00 JPY = 3,150.00

レコードナビゲートして更新します。

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)

(例)



※1つのタブが1つのテーブルに対応している。

【ポイント6】 業務機能を理解した上で開発手法を学ぶ

実装されている業務機能を知る

システム会社やITスキルの高い人によくあるケースであるが、実装されている業務機能を理解しないで、システムの開発手法(カスタマイズの手法)だけを学ぼうとするケースがある。

しかし開発手法だけ学ぶのは推奨しない。実装されている業務機能を知らないでカスタマイズの方法だけを学んでしまうと、既にある機能と同じような機能を実装してしまう無駄が発生し、全体的な開発工数が膨らんでしまう。さらに悪いケースになるとADempiere/iDempiereの開発標準に従わない開発(他のシステムの経験則での開発)が行われ、不用意なトラブルが発生してしまう。

実装されている業務機能は膨大なサンプル実装とも考えられる

開発者の立場でADempiere/iDempiereの業務機能を考えると、既に実装されている業務機能は膨大なサンプル実装とも言える。業務機能を知り、その実装方法を理解する事が、開発者にとって一番早くiDempiereの正しい開発手法が身に付くはずである。機能を追加開発する場合でも、既存の業務機能には参考となる所が多々ある。

少し手を加える事で使いやすさが格段に向上する

ADempiere/iDempiereは、ERPという性格上、機能が汎用的に作られており、オープンソースという特性上、ソースコードレベルでのカスタマイズを行う事もできるので、少し手を加える事で使いやすさが格段に向上する事が良くある。既存の業務機能に、どのように手を加えれば、導入企業にとって使いやすくなるのか把握しておく事は、SIを行う上では重要な事である。

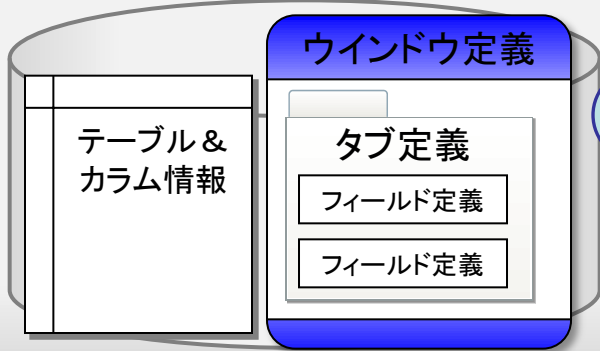
【開発手法】 GUI（ウィンドウとフォーム）の開発



ウィンドウ(Window) ⇒ 定型画面(標準画面)

- ウィンドウ(Window)はADempiere/iDempiere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/iDempiere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。

アプリケーション辞書



ウィンドウEngine

アプリケーション辞書の各種定義を読み込み、ウィンドウEngineにより自動でウィンドウ(定型画面)を作成する事ができる。

➤ バリデーションチェック
コールアウト(Call Out) Engine

“コールアウト”と呼ばれるプログラムにより入力値やフィールド間の整合性をチェックする事ができる。

Window 例

受注伝票 [Sales Order] SOO-1000015 SuperUser@株式会社OpenBiz.本社 [lo

ファイル(E) 編集(E) 表示(V) 移動(M) ツール(T) ウィンドウ W ヘルプ(H)

受注伝票 クライアント 株式会社OpenBiz 納品先 本社

受注伝票明細 伝票タイプ 標準受注

伝票番号 SOO-1000015

注文日付 2013/01/05

得意先注文番号

納品予定日 2013/01/05

説明

日取引先

取引先 サンプル得意先会社003

取引先住所 #1000008

取引先担当者

社内担当者 clientAdmin

日出荷

倉庫 本社倉庫

出荷ルール 在庫がある分出荷

配送方法 直送渡し

送料負担ルール 送料込み

直送販売

優先度 中

16000 - 3,000.00 - 合計: 3,150.00 JPY = 3,150.00

レコードをナビゲートして更新します。

1/3



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/iDempiere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。

フォーム Engine

クラス (Swing-UI用)

クラス (Web-UI用)

Form 例

出荷納品伝票作成 (マニユアル) SuperUser@株式会社OpenBiz.本社 [localhost:localhost-ide

ファイル(E) 表示(V) ツール(T) ウィンドウ W ヘルプ(H)

選択 / 作成

倉庫 本社倉庫

取引先

伝票タイプ

注文

伝票ステータス更新 [準]

組番	伝票タイプ	伝票番号	取引先	注文日付	明細行合計
☐	本社 Standard Order	SO1000002	サンプル顧客会社001	2012/12/23	8,000.00
☐	本社 Standard Order	SO1000003	サンプル顧客会社001	2012/12/23	3,000.00
☐	本社 Standard Order	SO1000009	サンプル顧客会社001	2012/12/27	13,000.00
☐	本社 Standard Order	SOO-1000019	サンプル顧客会社001	2013/01/06	5,000.00
☐	本社 Standard Order	SOO-1000016	サンプル顧客会社001	2013/01/06	3,000.00

受注伝票を選択して、出荷納品伝票を作成します。

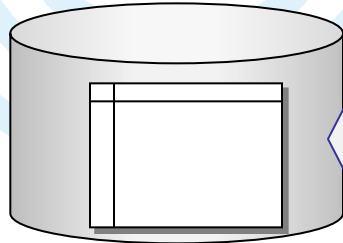
0

【開発手法】ビジネスロジックの実装

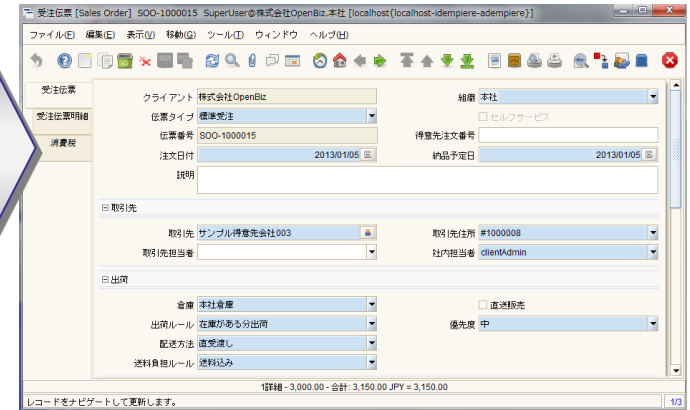
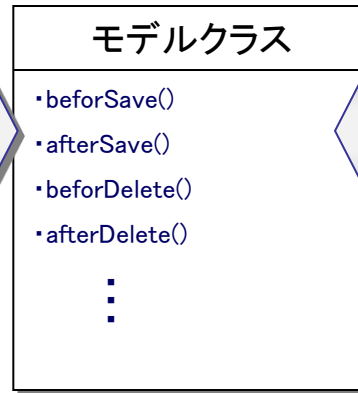


データの登録・更新・削除時のビジネスロジックの実装

- ADempiere/iDempiere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/iDempiere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/iDempiere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。

アプリケーション辞書

実行条件入力画面定義

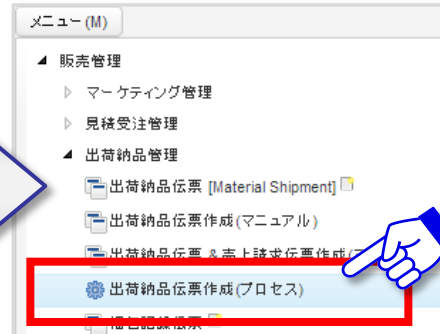
フィールド定義
フィールド定義



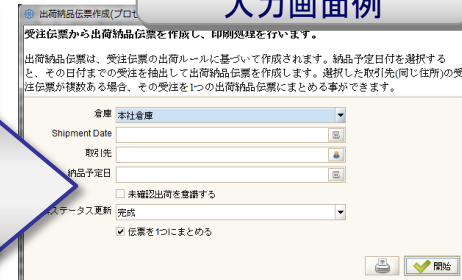
プロセス Engine

クラス

データ処理ロジックを記述



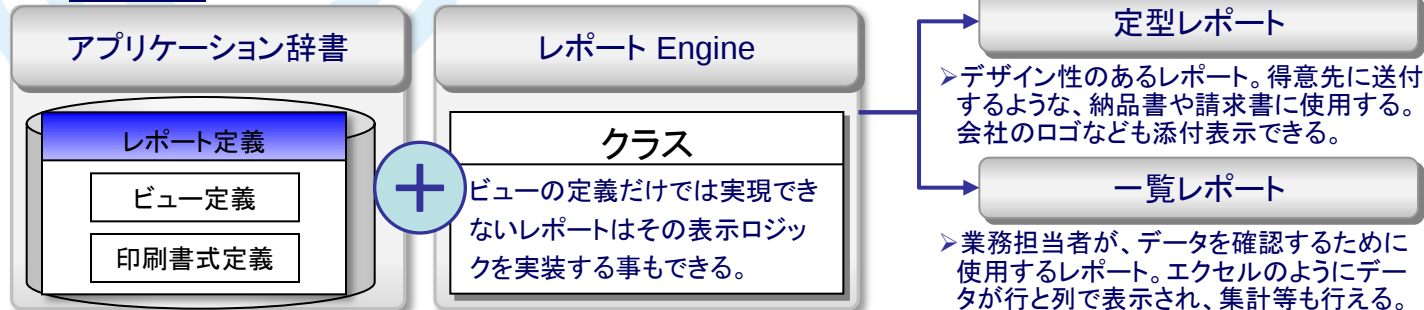
プロセス実行条件 入力画面例



【開発手法】レポートの開発

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

OpenStandia
請求書

請求日付: 2013/01/01
売上債権振替番号: A/R11000000

得意先: 関中
〒472-0000 (岐阜) 株式会社 OpenBiz
TEL: 017-0010102 FAX: 017-0010103

商品名: 部品A
数量: 10 単価: 2,000.00 金額: 20,000.00

20 SNAPL部品0001 5 5,000.00 15,000.00

30 SNAPL部品0002 2 5,000.00 10,000.00

9999 45,000.00 2,250.00

9999 47,250.00

備考: 請求書は、請求書として印刷されます。

印刷: 印刷済 印刷済 印刷済

代金振込先: 口座振替 口座振替 口座振替

ADempiere 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	説	明
100000	60000 - 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.	

会計レポート(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、管理会計の多次元分析のレポートが作成できるレポート作成ツール。

Jaspersoft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/iDempiere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/iDempiere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る。



ビジネスロジック
を中心に開発

```
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 > }  
254 > /** > Order Lines > > > */  
255 > private MOrderLine[] m_lines = null;  
256 > /** > Tax Lines > > > */  
257 > private MOrderTax[] m_taxes = null;  
258 > /** Force Creation of order > */  
259 > private boolean >> m_forceCreation = false;  
260 > private boolean >> m_forceCreation = false;  
261 > >  
262 > /**  
263 > * Overwrite Client/Org if required  
264 > * @param client_ID client  
265 > * @param org_ID org  
266 > */  
267 > public void setClientAndOrg (int client_ID, int org_ID)  
268 > {  
269 >     if (client_ID != 0 && org_ID != 0)  
270 >     {  
271 >         setClientAndOrg (client_ID, org_ID);  
272 >     }  
273 > }
```

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。

DB



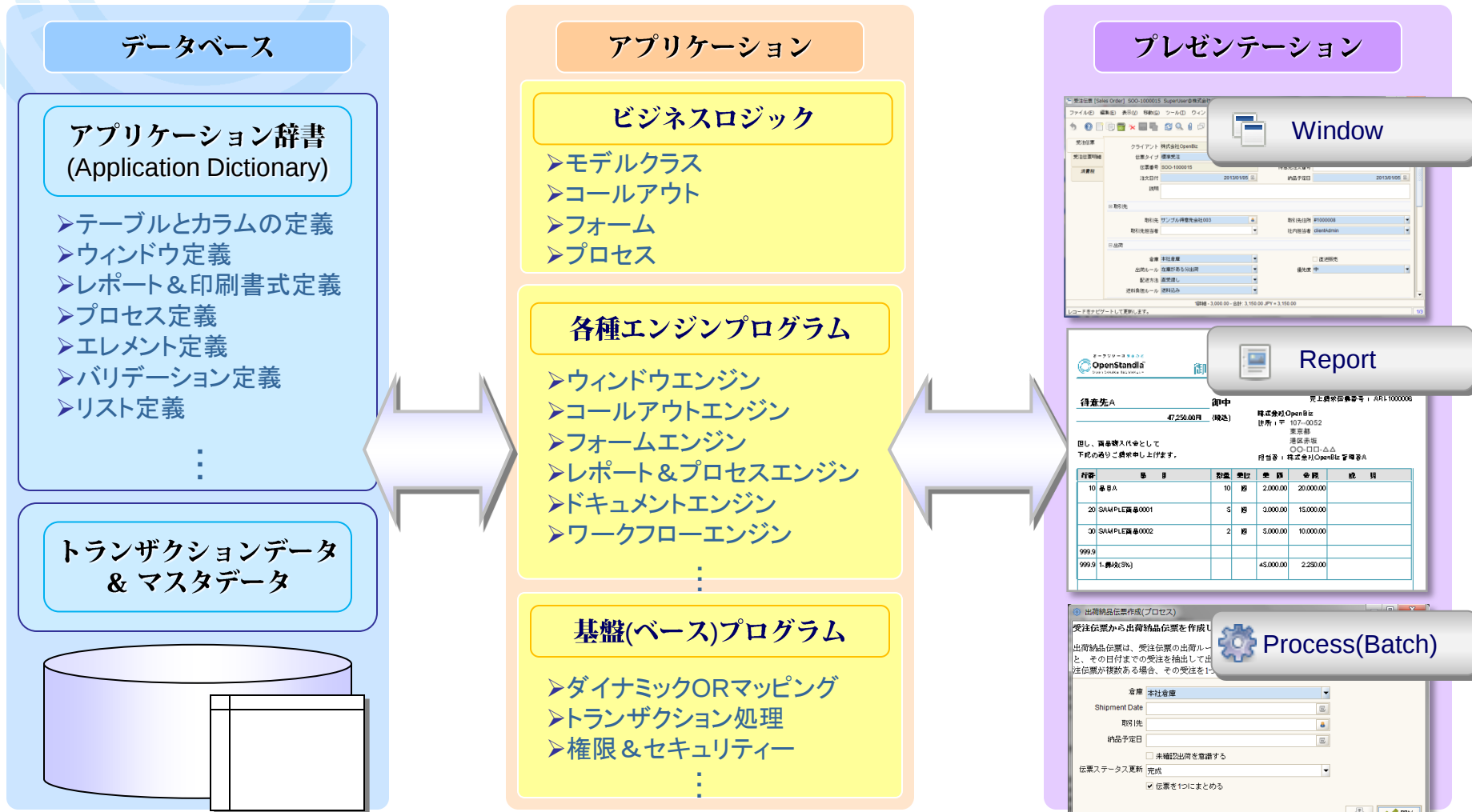
Process(Batch)

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。

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。

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



【ポイント7】自由にカスタマイズする

通常、商用ERPでは標準機能のソースコードを修正する事はできない。そのため、カスタマイズがパラメータ設定や、Add-Onプログラムの開発に限られる。しかしオープンソースなら、ソースコードが公開されているので、ソースコードレベルで自由にカスタマイズする事ができる。既存機能を上手に活用しつつ、足りない機能があったり、要件と機能にギャップがある場合には機能を追加開発してより自社の業務に適合した業務アプリケーションを作成する事ができる。

カスタマイズのコツ

既存機能を上手に活用する	既存機能を上手に活用する事が、開発工数の削減及びトラブル防止にもなる。
エンジンと呼ばれるプログラムを修正する際には、細心の注意を払う。	エンジンと呼ばれるプログラムは影響範囲が大きいので、モディフィケーションする際には、細心の注意が必要である。できれば、エンジンプログラムはモディフィケーションせずに導入する事を推奨する。iDempiereは通常、エンジンプログラムは修正する事無く導入する事ができる。
業務をシステムに合わせる事も検討する	自由にカスタマイズすると言っても、開発工数の関係上、やりたい事をすべて実装できるわけではない。時には、業務をシステムに合わせてもらう事も必要。業務をシステムに合わせる部分とカスタマイズする部分を区分する基準が大切。

カスタマイズの種類

Add-On (アドオン)	Modification (モディフィケーション)	Parameter / Configuration (パラメータ設定／コンフィグ設定)
既存のソースコードは修正せずに、機能を追加開発する事。	既存のソースコードを直接修正する事。	パラメータの設定で、カスタマイズする事。

【ポイント7】自由にカスタマイズする



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、**OSS-ERP(ADempiere/iDempiere)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。OSS-ERP(ADempiere/iDempiere)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OSS-ERP(ADempiere/iDempiere)のメリットを享受できていると言えるのではない!!
- 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のバージョンアップには通常、多額の費用が必要になるので、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OSS-ERP(ADempiere/iDempiere)はバージョンアップは気にせずに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。

【ポイント8】 導入に適している条件を知る

オープンソースの業務アプリケーションは自由にカスタマイズできるという特徴から、企業規模がある程度大きく、家電量販店で販売しているような既製品の業務パッケージソフトでは業務管理が難しい場合などに適している。実際の導入事例にはここで紹介している好条件には当てはまらない事例も多々あるが、導入を検討する上でのひとつの目安として参考にして頂きたい。

導入に適している条件(目安)を知る①

産業

カスタマイズ工数が一般的に少なく導入できると想定される産業を適合していると評価。

サービス業	◎	金融などは除く。
卸売業	◎	
小売業	○～◎	
製造業	△～◎	生産管理の導入形態による。
建設業	△～○	プロジェクト管理の機能強化(工事進行基準への対応など)が必要と思われる。
第一次産業	×	株式会社化されていれば検討の余地あり。

- 補足 -

上記の適合性の評価は、ADempiere/iDempiereをERPとして使用する場合であり、一部機能のみを使用するのであれば、その使用する機能によって適合性の評価も異なってくる。例えば、会計だけ使用するというのであれば、多くの産業で適合するであろう。

企業規模・形態

ADempiere/iDempiereが一般的に適しているとされる企業規模と企業形態。

- 中小企業(グループ経営している中小企業)
- 中堅企業(グループ経営している中堅企業)
- 大企業の子会社
- 海外展開している企業
- フランチャイズ展開している企業

年間売上高

5億円 ～ 500億円程度

従業員数

50名 ～ 500名程度

- 補足 -

上記の企業規模と企業形態の目安は、ADempiere/iDempiereを基幹業務システムとして全面的に使用する場合であり、適用業務範囲を限定する事により大企業でも利用する価値はある。例えば、ある新サービスの販売管理/債権管理などと限定すれば大企業でも十分利用できる。

【ポイント8】 導入に適している条件を知る



導入に適している条件(目安)を知る②

導入PJ期間

一般的なADempiere/iDempiereの導入プロジェクトの期間を評価。

3ヶ月未満	一部機能の使用でGapが少なければ検討する価値はあるが、ERPとして使用する場合は、導入期間が足りない恐れがある。	×～△
3ヶ月～半年	Gapがほとんどなく、コーディングを伴う追加開発がほとんど無ければ、3ヶ月程度で導入は可能(評価:△)。ある程度、期間に余裕を持たせ、少し手を加える事でADempiere/iDempiereは使いやすさが大きく変わってくるので、Gapが少ないケースは3ヶ月以上～半年程度の期間があれば評価は○～◎。	△～◎
半年～1年	機能に多少のGapがあっても、開発期間が充分にとれるので、ADempiere/iDempiereのメリットが発揮しやすい。	◎
1年～1.5年	導入企業が大きくなるとPJは長期間になる傾向があるため、1年～1.5年程度のプロジェクトとなるケースも珍しくはない。	○～◎
1.5年～2年	導入企業の規模が大きく、Gapも大きいと想定されるケース。	△～○
2年超	導入企業の規模が大きすぎるか、Gapが大きすぎる可能性があり、ADempiere/iDempiereのメリットよりリスクの方が大きくなる恐れがある。他のソフトと十分に比較検討する事が必要。グループ企業展開を含めて2年以上かかるというのであれば○。	×～△

導入PJ予算

一般的なADempiere/iDempiereの導入プロジェクトの予算感を評価。

100万円以下	自力で導入するケース。不用意なトラブルを回避するためにも、できれば導入経験やカスタマイズのノウハウがあるSI会社と一緒に導入プロジェクトを遂行する事をおススメする。	×
500万円前後	外部のコンサルタント2～3人月前後+αの費用。一部機能の使用で、Gapが少なければ○。ERPとしての使用を検討しているのであれば、Gapが少ない事が必要条件であり、場合によっては運用上ある程度の我慢や工夫が必要になってくる事がある。	△
500万円超～1000万円以下	中小企業でGapが少なければ○。	△～○
1000万円超～5000万円以下	機能に多少のGapがあっても、開発予算があるので、ADempiere/iDempiereのメリットが発揮しやすくなる。Gapが少なければ◎。	○
5000万円超～1億円以下	導入企業が大きくなると、PJの予算は大きくなる傾向があるため、5000万円超～1億円程度の規模になるケースも珍しくはない。	◎
1億円超～2億円前後	導入企業の規模が大きく、Gapも小さくはないと想定されるケース。開発工数が大きくなるとの恐れ、開発基盤としてのADempiere/iDempiereのメリットが享受しやすいケース。	○
3億円超～	導入企業の規模が大きすぎるか、Gapが大きすぎる可能性があり、ADempiere/iDempiereのメリットよりリスクの方が大きくなる恐れがある。他のソフトと十分に比較検討する事が必要。	△

【ポイント9】 OSS業務アプリのメリットとデメリットを理解する

OSS業務アプリケーションの特長とメリット&デメリットを理解する。

同じオープンソースでもLinuxなどのOSやPostgreSQLなどのミドルウェアは、通常ソースコードを修正するような事は無い。これらの分野のソフトは、オープンソースというよりも無料で使用できるという事が一般ユーザーにとっては意義が大きいだろう。しかし、業務アプリケーションは導入企業の要件に応じてカスタマイズが必要になってくる。ソースコードを自由に修正してカスタマイズができるというオープンソース本来のメリットは業務アプリケーションにこそ発揮される。

しかしその魅力の反面、カスタマイズが必要になるという事は、カスタマイズするための知識や経験が必要になるという事である。カスタマイズの方法を独学で習得する事もできるが、時間という貴重なコストを支払う事になる。

そして商業ソフトウェアと違い、通常開発コミュニティには、バグや不具合などに対して責任は無い。そのため、何らかのトラブルが発生した場合には、自己解決が求められる場合もある。

メリット	デメリット	備考
低コスト	開発コミュニティには不具合の責任は無い	・商用アプリケーションはどこまで責任を取ってくれるのだろうか!? ・OSS業務アプリを安心して使うためにはどうしたら良いのだろうか!?
不具合を自分達で修正する事ができる。	開発コミュニティが必ずしも不具合をすぐに修正するとは限らないので、場合によっては自分達で修正するか、修正できる人に依頼する必要がある。	不具合を自分達で修正できる事はメリットであるとも言えるし、デメリットであるとも言える。 しかし商用アプリケーションでも不具合を必ずしもすぐに修正してくれるとは限らないし、場合によっては仕様として押し切られて修正してもらえない事を考えると、自分達で直せる環境があるというのは良い事ではないか?
カスタマイズが自由	—	既存機能の熟知に加えて、カスタマイズのノウハウや経験値が重要!!
ベンダロックインされない	—	ベンダロックインとは、特定ベンダーの独自技術に大きく依存した製品、サービス、システム等を採用した際に、他ベンダーの提供する同種の製品、サービス、システム等への乗り換えが困難になる現象のこと。 [Wikipediaより]

【ポイント10】 困った時に頼れるサポート体制を構築する

困った時に頼れるサポート体制を構築しておく。

業務アプリケーションの場合、トラブルに迅速に対応できないと業務が滞ってしまい、大きな損失となりかねない。そのようなリスクを可能な限り抑えるために、困った時に頼れるサポート体制を予め構築しておく事も、オープンソースの業務アプリケーションを活用する上では大切な事である。

そして出来れば、サポート会社とは導入段階から一緒になって取り組むべきである。そうする事で標準機能の活用法やカスタマイズのノウハウを得る事ができ、不用意なトラブルを回避する事ができる。

オープンソースの業務アプリケーションはライセンス費用等が無料であるため、システム会社に導入を依頼しても、コスト的には商業ソフトウェアと比較して安価な場合がほとんどである。



参考サイト

iDempiere

Wiki	http://wiki.idempiere.org/wiki/Main_Page
WikiQSS:インストール及び開発環境構築の情報	http://www.globalqss.com/wiki/index.php/IDempiere
フォーラム/Googleグループ	https://groups.google.com/forum/?fromgroups#!forum/idempiere
バグチケット管理/JIRA	http://jira.idempiere.com/secure/Dashboard.jspa
ソースコード管理/bitbucket	https://bitbucket.org/idempiere/idempiere
ビルド(CI)/jenkins	http://jenkins.idempiere.com/
Sourceforge	http://sourceforge.net/projects/idempiere/

ADempiere

Wiki	http://www.adempiere.com/ADempiere_ERP
フォーラム	http://www.walkingtree.in/forums/forumdisplay.php?11-Adempiere
日本ADempiereの会(JADG)	http://adempiere-jp.net/
ADempiere.jp	http://www.adempiere.jp/
Sourceforge	http://sourceforge.net/projects/adempiere/

- OpenStandiaは、「攻めのIT」を支援します。
- オープンソースのことなら、なんでもご相談ください！



お問い合わせは、NRIオープンソースソリューション推進室へ



osscc@nri.co.jp



<http://openstandia.jp/>