

第2回インターネット時代のCOBOL活用セミナー



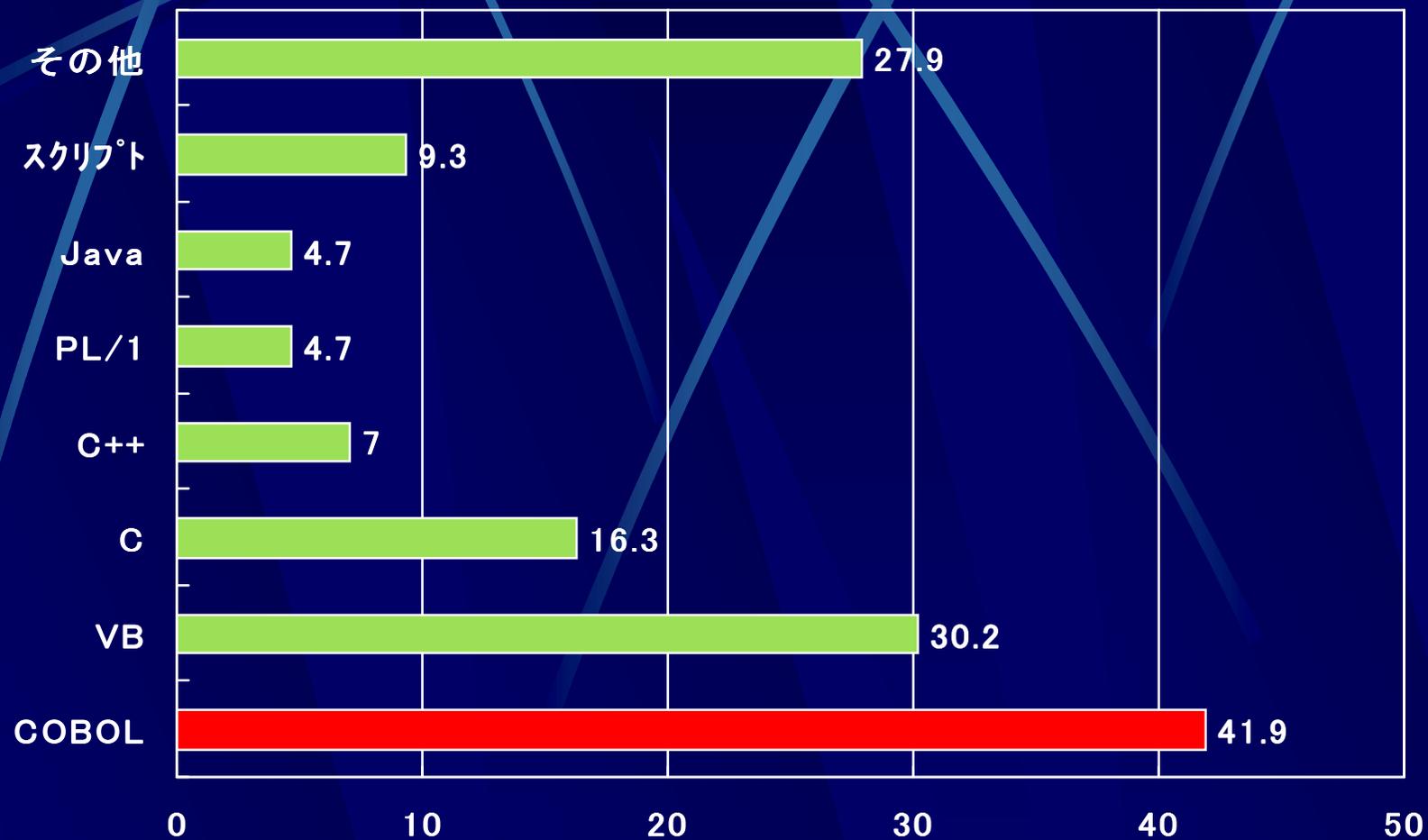
# COBOL資産活用 ～メリット、モデル、留意点～

COBOLコンソーシアム セミナー分科会 幹事

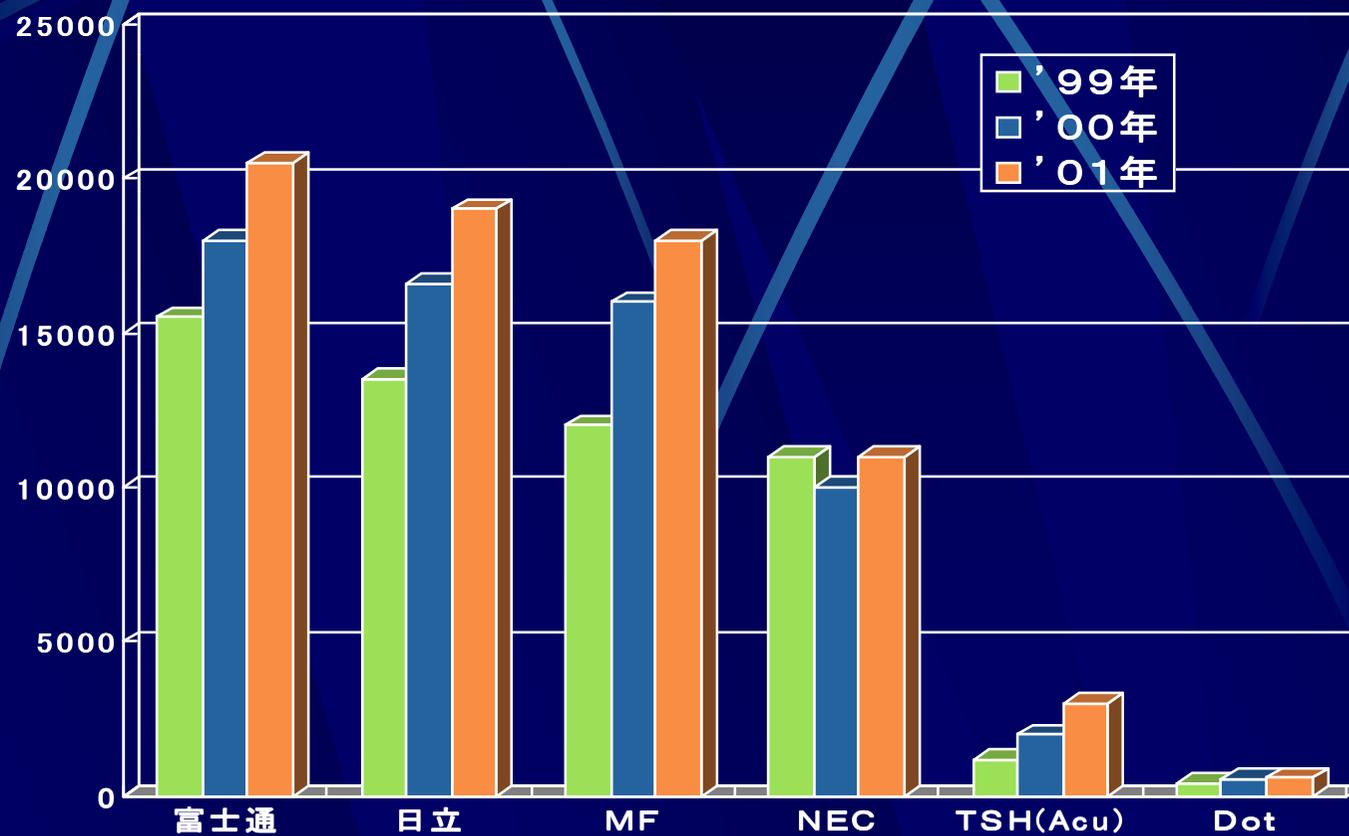
東京システムハウス株式会社

林 知之

# 新規開発言語 ('00年日経BP:首都圏上場企業)



# COBOL出荷状況 ('01.11日経システムプロバイダ)



# COBOLの資産価値

- **COBOLソースコード**
  - 数億本？（米国では、1億8千万）
  - 大企業では1社で数千本～数万本
  - 金額では、数千億円
- **技術者（SE、プログラマ）**
  - 全世界で200から300万人(※)
- **既存システムの70%がCOBOLで稼動中(※)**

※Giga Information Group, May 2001

# COBOL資産活用のメリット

- 高い生産性/保守性
- 共通性/標準規格
- 確立された開発技法
- 業務知識を豊富に有した技術者
- 事務処理に適した言語仕様
- 先端技術に対応
  - RDB、GUI、分散オブジェクト、他言語I/F、
  - Web(CGI、Plug-In、スクリプト、Webサービス、etc)
  - サーバーベースドコンピューティング

# COBOL is Business

- 40年以上蓄積された資産
  - これだけの実績(歴史)をもつ言語は、COBOLのみ
  - 数多くの開発言語、ツールが誕生して、消えた
- COBOL資産は、ビジネス資産そのもの
- 企業は、自己のビジネスの利益拡大が目的
- 既に有るビジネス資産を活用すべき
- バブル経済、IT・ネットバブルの崩壊
- 本来のビジネスを効率良く拡大させる為のIT



# COBOL資産活用モデル

# COBOL資産活用モデルとその事例

1. ダウンサイジング
2. パッケージビジネス
3. RDB連携とEUCの実現
4. ASPサービス
5. E-Business&Mobile展開
6. サーバベースドアプリケーション
7. ホスト連携モデル

# ダウンサイジング(TCO削減)

## 【費用効果】

項目	旧システム	新システム	コスト削減率
ハード	汎用機	Unixサーバ1台 Unixワークステーション1台 Windowsサーバ1台	70%減
	専用端末機	既存Windows端末をそのまま利用	新規投資無
ソフトウェア	汎用機で導入しているソフト使用料が毎年発生	COBOL SORTツール 運用管理 帳票 etc	購入ソフトを5年回収とすると50%減

# ダウンサイジング (TCOの削減)

## 【ランニングコスト効果】

項目	内容	削減効果
リース料	汎用機リース料と新システム設備投資リース料との比較	60%以上減
通信費	専用線によるセンターとの通信をTCP/IPによる通信へ切り替え	年間300万円減
保守料	ハード・ソフト保守料	60%以上減
電力料	汎用機使用電力、エアコン etc	年間15万kw 200万円減
スペース	汎用機本体、MT保管場所	約1/5以下に削減

機能面での向上によりコスト削減にも貢献している。  
夜間以外に行われるバッチ処理の処理時間は、5~10倍の処理速度の改善により、オペレータの人件費が、そのまま1/5~1/10へと削減される。

# パッケージビジネス

## 【事例】

もともと、オフコンで稼動していたパッケージシステムをPC版として、DOS上に移行。その後、Windowsの登場により、GUI画面の必要性(ニーズ)が高まった。

システムの改善を含め  
新システムの構築を検討



他言語(COBOL以外も含め)1年  
半検討・評価

ACUCOBOL  
との出会い

選定条件を全て満たす  
COBOLを採用

### 選定条件

- ・GUI画面の採用
- ・Web対応(将来性)
- ・開発スピード・コスト(生産性)
- ・1社のOSの依存度を低める

# パッケージビジネス

COBOL: 約1000本(プログラムステップ: 約100万ステップ以上)  
開発着手してから、約8ヶ月でテストユーザへの導入、10ヶ月で正式出荷



開発技術ノウハウの継承  
現行プログラムの流用  
他言語との容易なる連携(一部VBを採用)

- ・開発コストの削減 = 既存資産の流用、早く安く正確に移植
- ・価格競争力の向上 = 低価格なCOBOLランタイム
- ・保守コストの削減 = リソースの一元管理(OS毎の管理が不要)  
⇒ 修正・変更時の生産性向上、使い慣れた言語での保守
- ・マーケットの拡大 = Windows稼動環境によるマーケット拡大
- ・競合製品との差別化

# RDB連携とEUC実現

## 【事例】

高機能パソコン&Office環境が整備され、基幹システムに蓄積されたデータの活用ニーズが高まってきたが、既存のシステム環境では、対応が困難となってきた。従来は、エンドユーザからの要望された帳票を作成するまで、数ヶ月かかっていたが、近年の情報化社会において、即時対応出来なければ、企業が生き残れなくなっている。また、個別(部署・課・個人)のニーズへの対応が増えている。

### その他問題点

1. オフコン製造の打ち切り
2. エンドユーザの情報リテラシーの高まり
3. グループウェアとの連携ニーズ

COBOLでのマイグレーション  
ODBCでのEUCの実現

# RDB連携とEUC実現



- ・業務システムで蓄積したデータの活用が容易になる。  
⇒各個人の業務にあった分析が、即時可能(経営判断に直結)
- ・拠点及び社外でのデータ活用が可能になる。  
⇒情報の共有化
- ・個別帳票の削減による保守メンテナンスの向上  
⇒個別対応の作業の減少。(メンテナンスコストの削減)  
⇒新技術研究時間の増大。(戦略的な情報システム企画)

# ASPサービス

## 【事例】

情報化社会による新しい事業の展開として、ASP事業の検討  
今までの経験や技術を活用したパッケージを選定

### 【パッケージ開発の条件】

- ・Linuxを採用
- ・インターネットでの提供
- ・COBOL経験者による生産性の向上
- ・メーカ依存度の削減
- ・柔軟性と将来性が見込める



## COBOLを採用

## 構成

### 【Server】

O S : TurboLinux Server 6.1  
D B : PostgreSQL 7.0.3  
開発 : AcuCOBOL 4.3.1  
(cgi および C/S方式)  
Web : apache 1.3.14  
その他 : HDD ミラーリング  
PDF作成ツール採用



Internet

### 【Client】必要ソフト

ブラウザ  
AcrobatReader  
プラグイン  
SQLランタイム  
(上記2製品は一部使用)

# ASPサービス

## 【開発側の効果】

- ・COBOL経験者によるWebアプリケーションの構築が可能
  - ⇒リリースまでの開発期間の短縮
  - ⇒開発コストの削減
  - ⇒高品質なアプリケーション構築
  - ⇒メンテナンスコストの削減
- ・異なるプラットフォームへの移植が可能
  - ⇒開発環境は、Windows、本番環境は、Linux
  - ⇒初期コストの削減



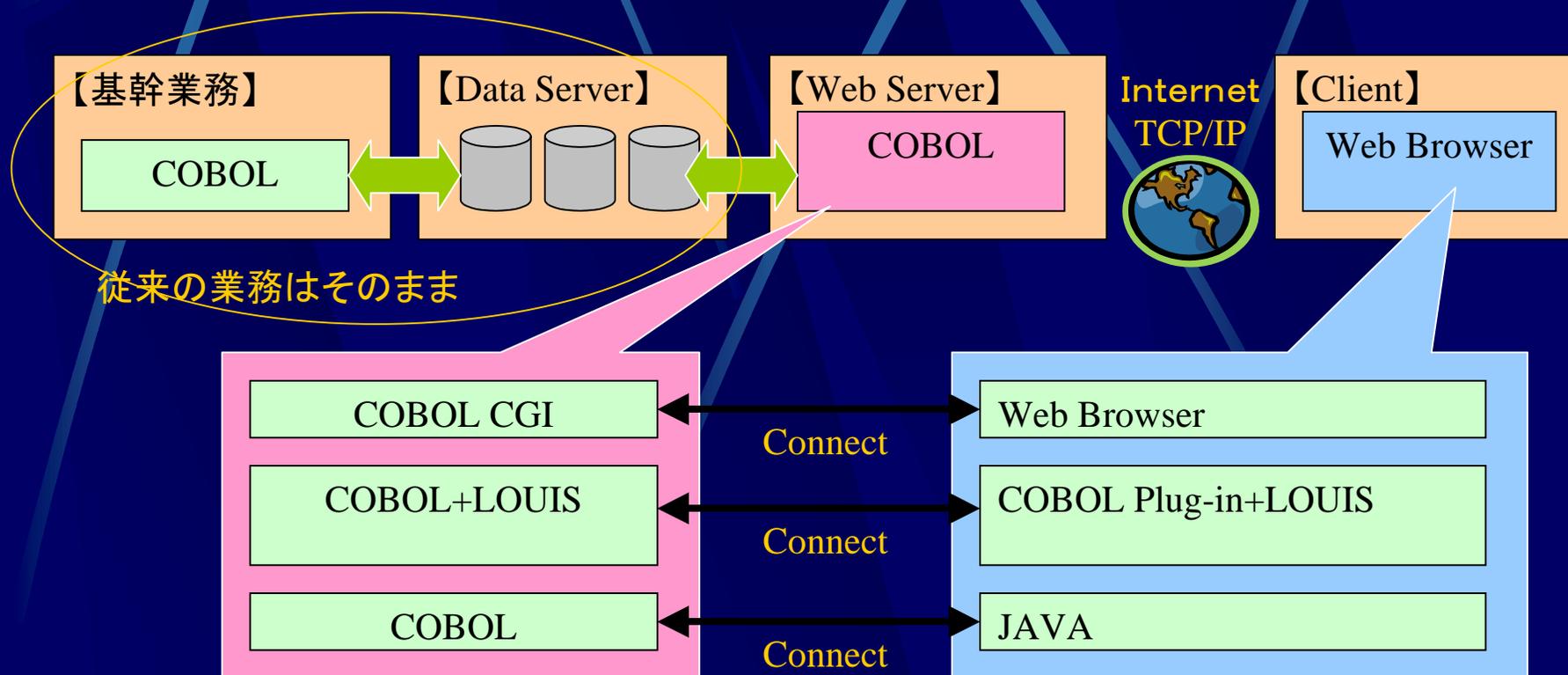
## 【ユーザ側の評価】

- ・INTERNET接続環境があれば利用可能
  - ・迅速な仕様改定への対応
  - ・サーバ管理不要
  - ・業務に合わせたカスタマイズ対応
  - ・販売形態の選定  
(ASP、パッケージ販売、レンタル)
  - ・無償バージョンアップ
- ASPの特徴
- 販売元のサービス

# E-Business & Mobile

## 【ご提案】

携帯電話やPDAの普及に伴ない、外出先からの情報収集が容易になってきた。それらを基幹業務のデータと接続することで、主業務の効率化を図ることが可能となる。



# E-Business & Mobile 展開



基幹業務

通信速度  
セキュリティ

- 情報収集の強化
- 情報の共有
- 業務の効率化
- 事業の拡大
- 新事業の発展

主業務の商売に大きく貢献

守りの情報システムから  
攻める情報システムへ

# サーバベースド・アプリケーション (MetaFrame, Thin Client)

**【ご提案】** オープン環境でホスト集中型管理を可能  
クライアントには、コンパクトなUIモジュールのみ

今までの、ホスト・オフコンのシステム構成そのまま、オープン環境に移植。

COBOL構造を変えないで実現



# サーバベースド・アプリケーション (MetaFrame, Thin Client)

## 【TCOの削減】

- ⇒開発コスト
- ⇒初期コスト
- ⇒運用コスト
- ⇒最新技術の取り込み

今までのマイグレーション  
より更に効果を発します

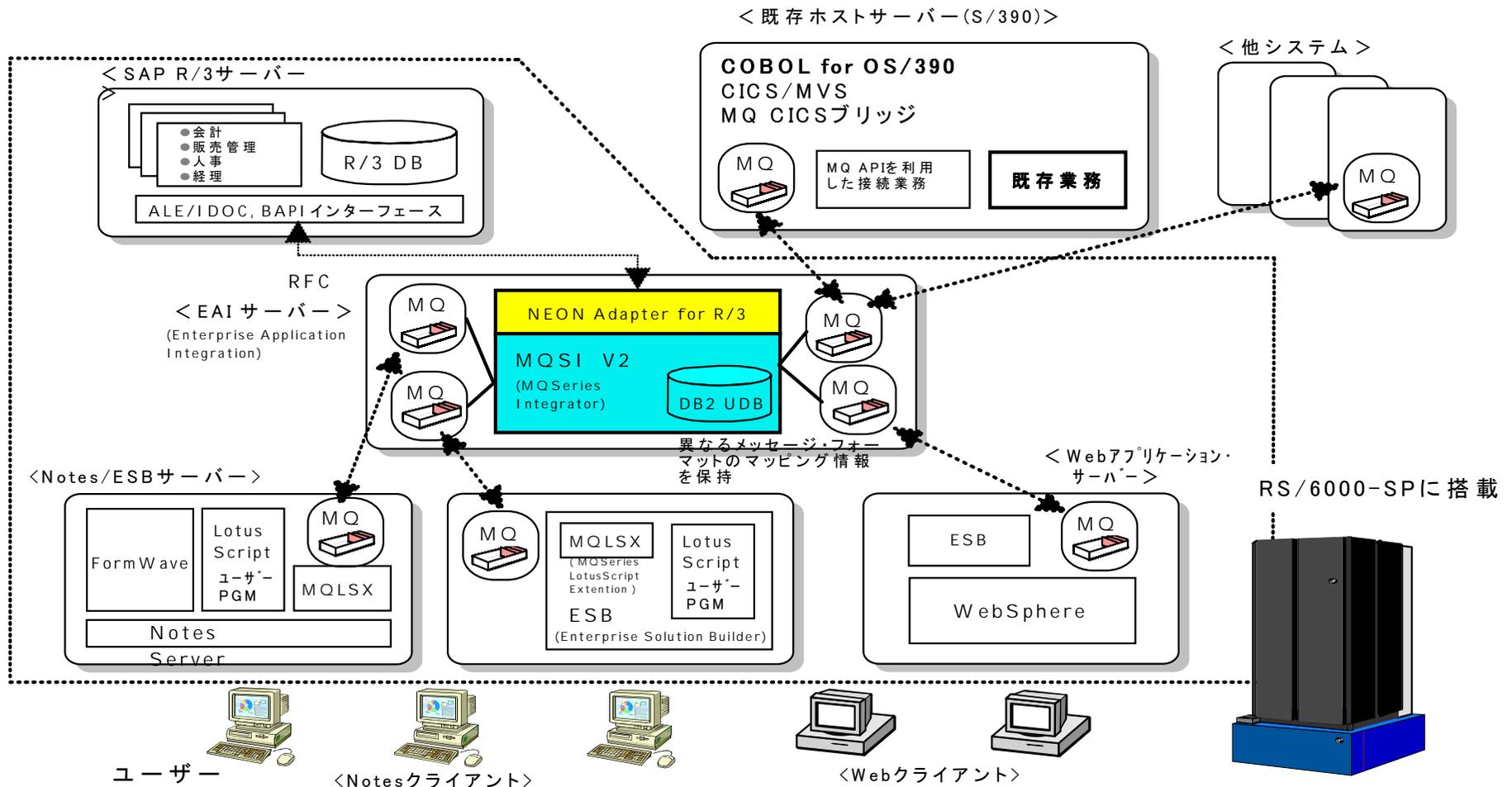
COBOLの先進性をそのま  
ま利用できます

今までのマイグレーション効果を更に磨きを掛け、“安く・早く・正確に”オープン環境へ移行することが実現できます。

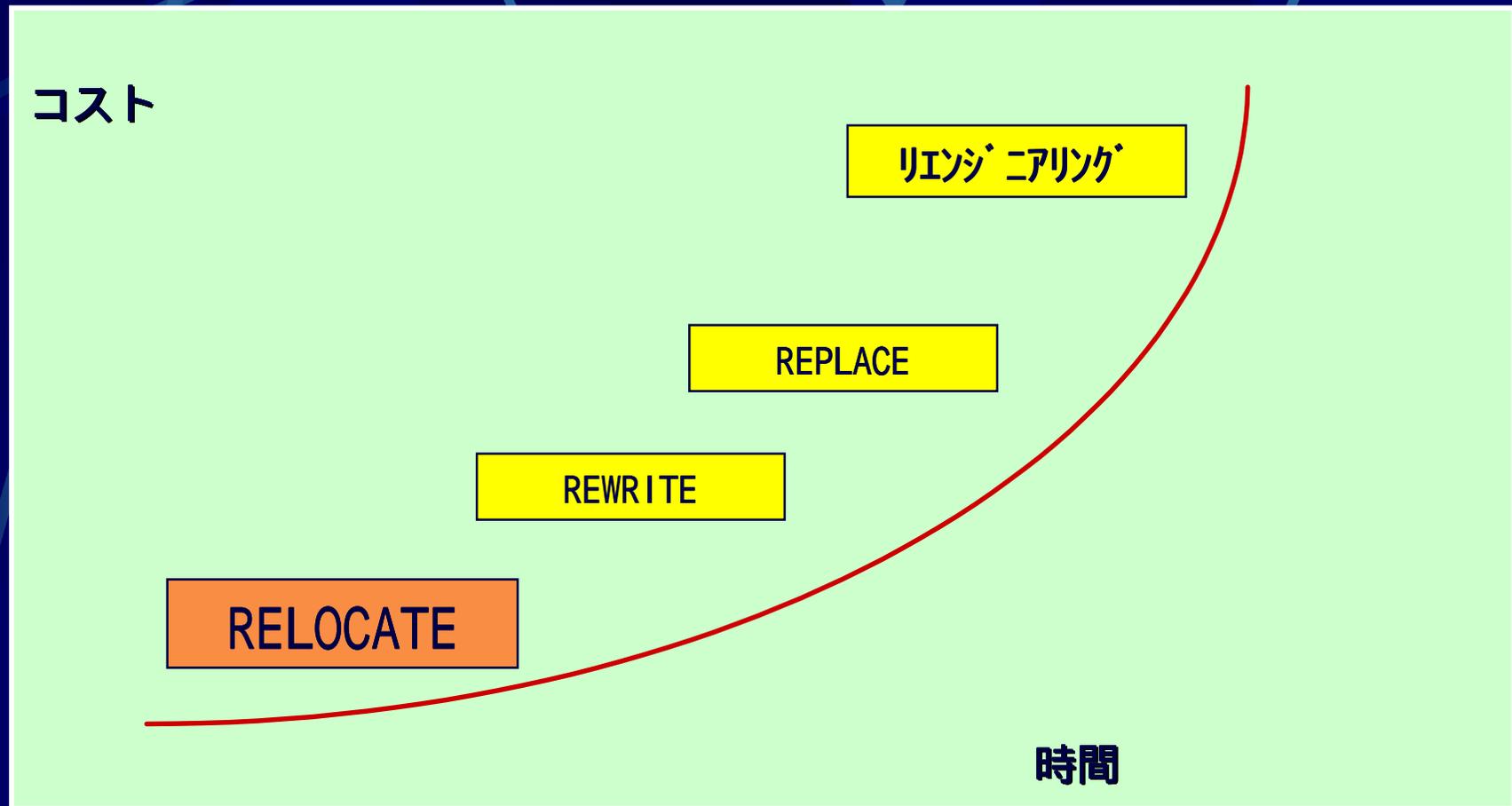
# ホスト連携モデル

## 既存COBOLの活用例 - 既存業務、SAP R/3とNotes/Web連携ソリューション

EAIサーバー内にMQSeries Integratorを置くことにより、各サーバー間で必要なデータのマッピングと転送をおこなうことができます。



# COBOL資産の活用は



# COBOL資産活用の留意点

- COBOL資産の定義
- オープンシステムへの移行手順
- 移行時の留意点

# COBOL資産の定義

- COBOLプログラム
- JCL
- 画面
  - 定義ツール、オンライン
- 帳票
  - オーバレイ、高速NIP、分散プリント、電子帳票
- データ
  - SAM,ISAM,VSAM,DB(階層型、関係型等)
  - MT(オープン、カートリッジ、FD?)

# COBOL資産の定義

- システムライブラリ(各種ツール)
- 簡易言語
  - Mapper,NHELP,SMART 他多数...**使いすぎ!!**
- 運用管理ツール
- OLTP
- システム管理保守関連ツール
- テスト関連ツール

# オープンシステムへの移行手順

- 既存システムの調査・分析
- 移行先システムの決定
  - プラットホーム、Web、DBMS、GUI、EUC
  - 各種機能(帳票、OLTP、運用管理、etc)
- リソースのコンバージョン
  - COBOL、JCL、画面定義、帳票定義
  - 運用関連、オンライン定義、システム定義
- ハードウェア選定
- テスト、ユーザー教育、導入へ

# 移行時の留意点

- COBOL to COBOLでもそのまま使えない
  - 文法、動作(同じ命令でも)、算術の中間桁数  
ファイルI/Oのステータスチェックなど
    - 挙げればきりが無いが、コンパイロプションの設定  
や、自動変換ツールで対応出来る部分が多い
- JCL
  - 特にホストは不要な部分が多い
  - 但し、手作業による変換作業が多い

# 移行時の留意点

## ● 画面定義等

- 変換ツールを利用可能（CUIが基本）
- GUIへの変換や、画面定義はワークベンチ等で容易に出来るようになって来た
- CGI、HTML、Java連携、他言語連携
- サードベンダーツール、サーバーベースコンピューティング
- 利用場面毎のユーザーインターフェイス
  - デスクトップ、ノート、モバイル、PDA、携帯電話

# 移行時の留意点

## ● 帳票関連

- オーバレイ
  - 各メーカーなどで変換サービスも実施
- 大量印刷
- プリントキュー(再印刷など)
- 分散印刷
- 専用用紙(複写)
- 電子帳票

※目的に応じて検討が必要！！

# 移行時の留意点

- データ移行
  - マスタ、トランザクション
  - ディスク、MT
  - SAM、ISAM、階層型DB、関係型DB、他
  - コード変換
    - 漢字、EBCDIC、SJIS、ASCII、EUC
    - Sortの順序
  - 移行のタイミング(本番時は一気に！)

# 移行時の留意点

- システムライブラリ
  - 不要なものが多い
  - 新環境で代替機能作成
    - 既に多くのツールが開発されている
- 簡易言語（これが厄介！！）
  - 殆んどは作り変え
    - 但し目的に合わせて、より簡単に
  - 一部のツールは変換可能
- 簡単だからといって先を考えずに使いすぎるな！！（適用場面を間違えると後が大変）

# 移行時の留意点

- 運用管理ツール
  - OLTP
  - システム管理保守関連ツール
  - テスト関連ツール
- 
- その他既存機能、新環境でのツールの選定、採用も重要なポイントとなる。

# 移行に当たっては

- 経験豊富なシステムベンダーに是非ご相談ください

ありがとうございました。