

COBOL資産を活用した 富士通の間接材購買システム統合事例

2008年 9月5日
富士通CIT株式会社
ITシステムグループ プロジェクト部長
亀田 斉司

Copyright 2008 FUJITSU CIT LIMITED

富士通CIT

■ 事業内容

富士通CIT株式会社 (FJCIT) は、SCMをコアとしたものづくり革新を支えるITソリューションに加え、インフラ・ネットワーク構築から情報セキュリティ管理までを含めた基幹情報システム全般について、調査・企画・開発・運用のライフサイクルマネジメントを追求していきます。

■ 本社

〒323-0807 栃木県小山市城東3-28-1 (富士通小山工場)
電話 0285-24-6319

■ 資本金 2億5千万円 (富士通株式会社 全額出資)

■ 売上高 104億3200万円 (2008年3月期)

■ 社員数 660名 (2008年4月1日現在)

1. 間接材購買における コスト削減への取り組み

間接材購買とは

- **事務用品などの総務購買**
 - 什器、文房具、名刺、サプライ品 など
- **通信キャリアへの支払を中心としたネットワーク購買**
 - 開発用の専用線、携帯電話、データ通信用カード など
- **設備のメンテナンスなどの建設購買**
 - 事務所内のレイアウト変更、補修工事 など
- **その他**
 - 広告費の支払いを中心とした宣伝購買、各種サービスの利用費

間接材の購入実績は1,500億円以上(2004年度単独実績)

プロジェクトの開始

FUJITSU

これまでの間接材購買の課題

- 調達というより「固定的な費用の支払」という意識が散見
- 費用削減活動は各部門ごとにバラバラに進められていた

プロジェクトを発足し、コスト削減活動を開始

調達部門の観点から削減余地を検討し、最適な調達を実践

【活動ポイント】 競合／合見積の実施、価格交渉の実施
グループ間での情報共有(ボリューム交渉の実施)

CITとしてこれら活動をITで支援

4

Copyright 2008 FUJITSU CIT LIMITED

間接材購買業務の問題点1

FUJITSU

購買情報が価格交渉や経営判断など、戦略的に活用できていない

理由

購買情報が16部門に点在し、購買金額の実態把握が困難



5

Copyright 2008 FUJITSU CIT LIMITED

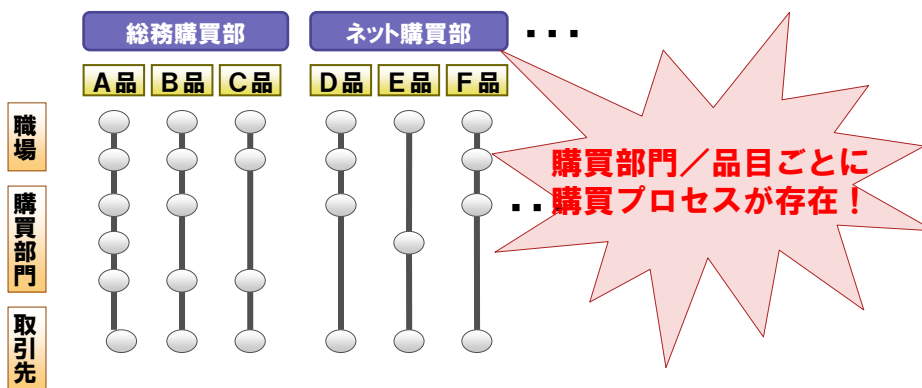
間接材購買業務の問題点2

FUJITSU

同種の間接材を複数部門が独自ルールにて購入していて非効率

理由

購買部門が複数あり、業務プロセスが統一されていない



6

Copyright 2008 FUJITSU CIT LIMITED

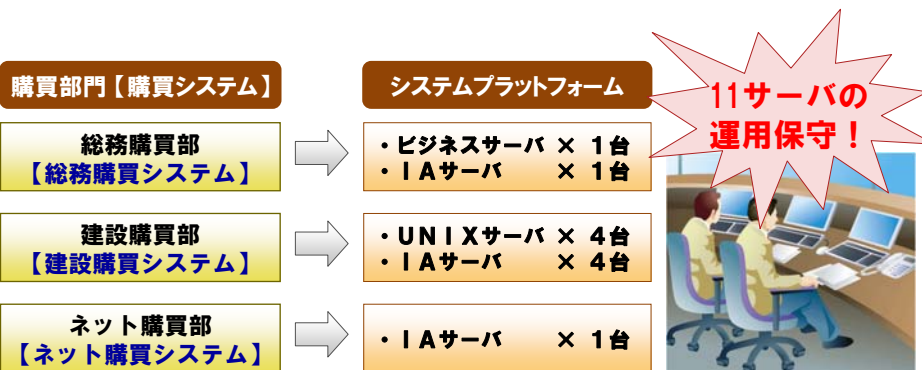
間接材購買業務の問題点3

FUJITSU

購買関連システムの運用コストが適正化されていない

理由

部門で購買システムを構築し、個々に維持/運用が行われている



7

Copyright 2008 FUJITSU CIT LIMITED

解決アプローチ

FUJITSU

問題	原因	対策
1) 購買情報が戦略的に活用できていない	- 購買情報(実績)が部門に散在	- 購買情報の集約 (データの一元管理化)
2) 購買情報が戦略的に活用できていない	- 購買部門が複数あり、業務プロセスが統一されていない	- 部門独自の業務フローの統一
3) システム運用コストが適正化されていない	- 部門独自のシステム構築・運用を実施	- 部門独自の購買関連システムの統合

8

Copyright 2008 FUJITSU CIT LIMITED

具体的な施策

FUJITSU

Step1: 購買情報の見える化

- 全社の購買実績の一元管理化
- 取引先別の購入金額など、分析ビューの提供

購買情報DWHシステムの構築

Step2: 購買関連システムの統合

- 全社で36プロセスある業務フローの標準化
- 散在していた購買関連システムを統合して1本化

間接材購買システムの構築

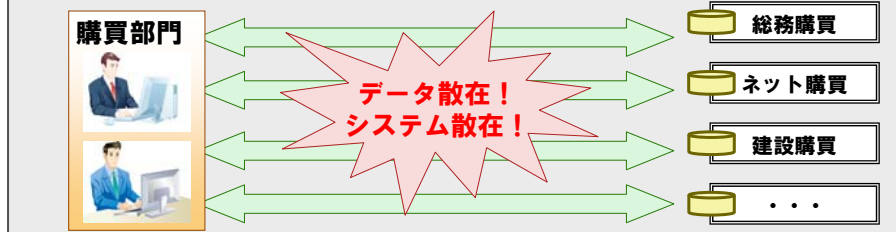
9

Copyright 2008 FUJITSU CIT LIMITED

施策の全体イメージ

FUJITSU

従来



今後



10

Copyright 2008 FUJITSU CIT LIMITED

プロジェクト目標

FUJITSU

間接材購買におけるコスト削減 (目標: 10億円/年)

コスト削減の3本柱

- (1) 調達コストの削減
- (2) 調達総量の削減

→ 購買情報の見える化により、スケールメリットなどを生かした価格交渉にてコスト削減

- (3) システム運用コストの削減

→ 購買関連システムの統合にてTCOを削減

11

Copyright 2008 FUJITSU CIT LIMITED

2. 購買関連システムの統合

システム構築のポイント

業務標準化の徹底

- 各購買部門の業務プロセス全ての実装は工数増
- システム統合は業務プロセスの標準化から

既存資産の有効活用

開発にはマイグレーションを適用し、開発工数／期間を短縮

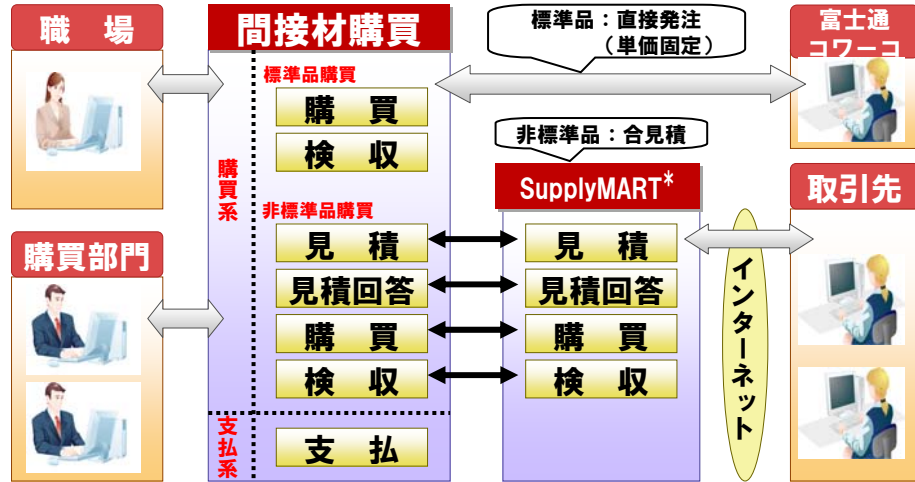
新規技術の積極採用

- IA基幹サーバ(PRIMEQUEST)の採用
- 基幹Linuxによる基幹系システムの構築

システムイメージ



標準品は直接発注、非標準品は合見積の仕組み

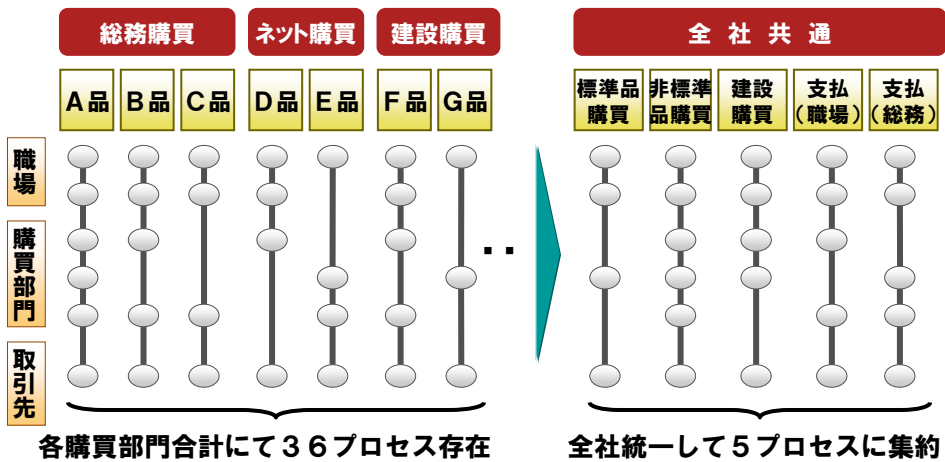


*: オフィス用品等の調達を行うインターネット上の共同マーケット
Copyright 2008 FUJITSU CIT LIMITED

業務プロセスの標準化



購買種別を分類し、5プロセスへ標準化

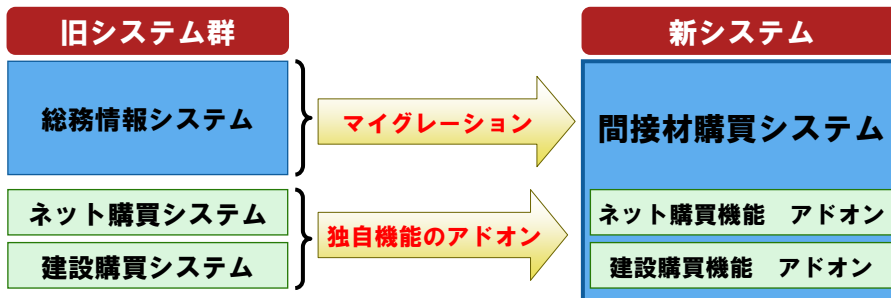


マイグレーションによるシステム構築

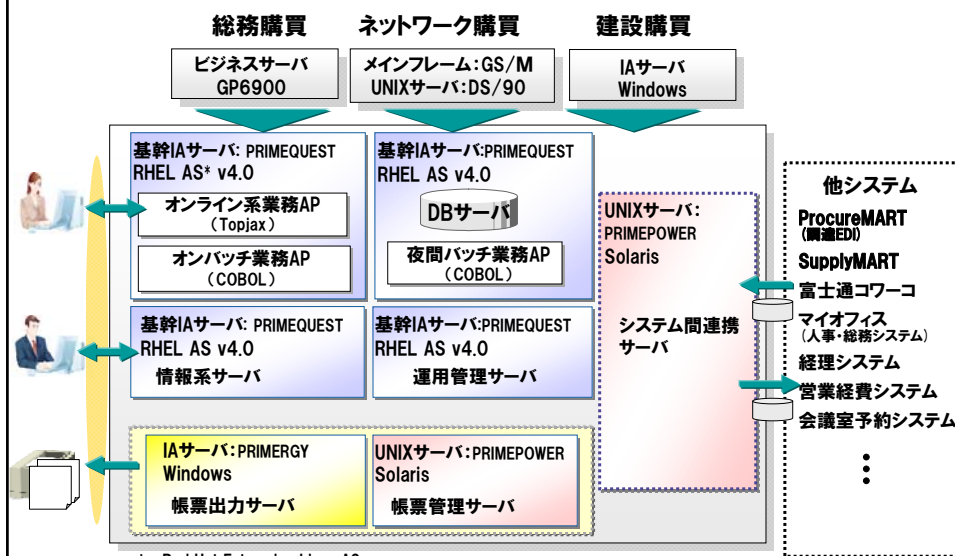


既存資産の活用のためマイグレーションを適用

- 開発費用の抑制、スケジュールの短縮
(今回のプロジェクトではスケジュール約6ヶ月の短縮)
- リスク軽減(正常に動作していた資産の流用)



システム構成概観



開発プロセス概要



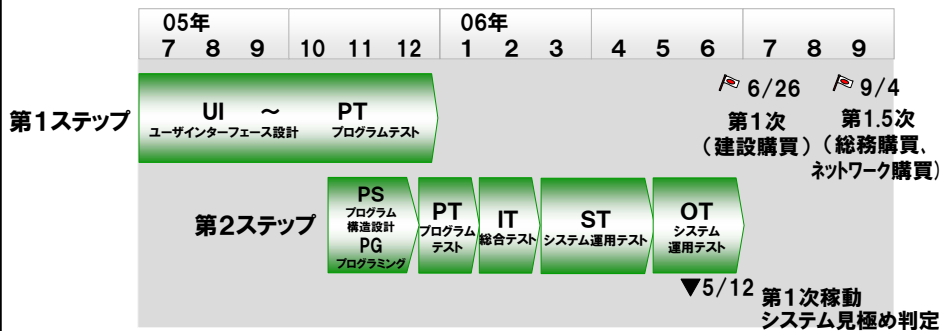
■ 二段階でGOALの基幹IAサーバ(PRIMEQUEST)に移行

第1ステップ： 現行プラットフォームからUNIXサーバに資産移行

- オンライン・アプリケーション：Topjax*による新規作成（リビルド）
- バッチCOBOLアプリケーション：NetCOBOLでリライト、プラットフォーム、アーキテクチャ相違の対応

第2ステップ： UNIXサーバから基幹IAサーバ(PRIMEQUEST)に資産移行

- オンライン・アプリケーション：マルチフォーマット対応のみ
 - バッチCOBOLアプリケーション：Unicodeにフォーカスした対応
- *：補足ページ参照



移行資産・規模

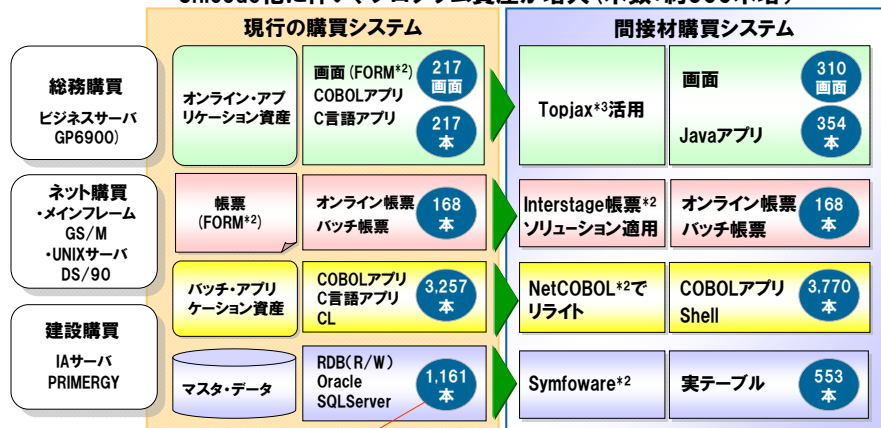


■ 実績

オンライン：3つの購買機能の統廃合により規模拡大も初期投資どおり ⇒ 生産性1.5倍

バッチ：ビジネスサーバのCL*1を「Shell+COBOLアプリ」で実現。

Unicode化に伴い、プログラム資産が増大(本数:約500本増)



ファイルと実テーブルの合計

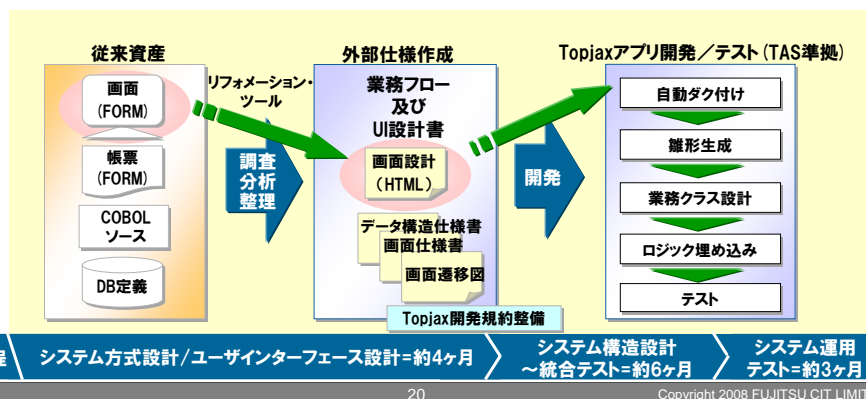
*1:制御言語、*2:富士通のミドルウェア製品 *3:補足ページ参照

オンライン・アプリケーション構築

FUJITSU

■ Topjaxによるアプリケーション開発を実施

- プログラムと仕様書(データ構造仕様書、画面仕様書、画面遷移図)の同期化と、Topjaxの標準化された「わかりやすいアプリ構造」により、メンテナンスビリティを大幅改善
 - Java開発未経験者もTopjaxの短期教育で標準レベルのアプリケーション開発スキルを習得させ、開発コスト低減化を実現
- ⇒ 高度な開発技術を必要とせず、短期で開発体制を構築(ピーク時約60名の開発体制を確保)



20

Copyright 2008 FUJITSU CIT LIMITED

補足: Topjaxとは

FUJITSU

Total Omni Platform architecture by java and Xml technology

Webシステム構築において先進技術 (Interstage^{*}、XML、Java、ブレードサーバ等) を採用し、上流工程から下流工程までを体系的、かつ効率的にシステムインテグレーションするための
プロダクト群とフレームワーク

セミオーダーメイド SIモデルによる
システムインテグレーションとワンストップソリューション
により、開発期間短縮を実現

*:富士通のミドルウェア製品

21

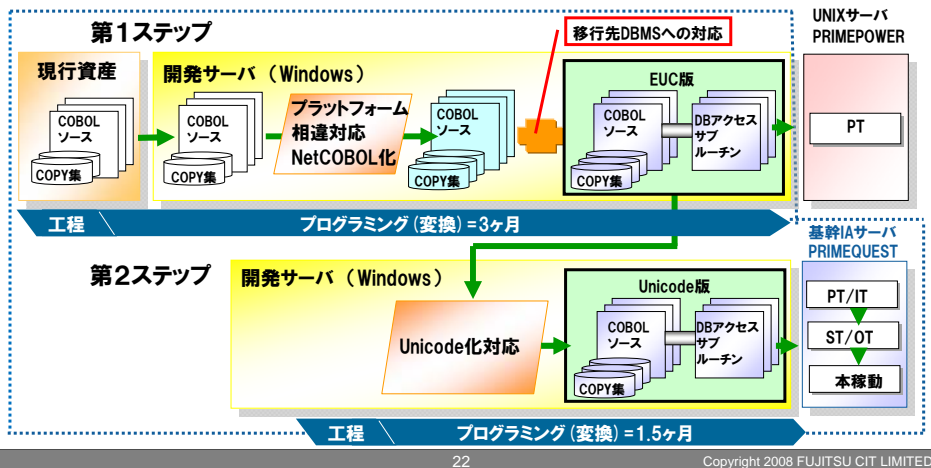
Copyright 2008 FUJITSU CIT LIMITED

バッチ・アプリケーション構築

■ 二段階でGOALのIA基幹サーバ(PRIMEQUEST)に移行

第1ステップ： 現行プラットフォームからUNIXサーバに資産移行

第2ステップ： UNIXサーバからIA基幹サーバに資産移行



3. マイグレーションのポイント

COBOL資産移行を成功させるには



■ COBOL to COBOLベースでの資産移行の光と影

➤ 他言語によるスクラップ&リビルドと比べたCOBOL to COBOLのメリット

- 開発期間が短い
- 低コストである
- 現行の安定性を引き継げる
- プログラムや人材を活かせる

光

➤ COBOL to COBOLの注意点

影

- COBOLコンパイラの相違 (メインフレームCOBOL → オープンCOBOL)
- プラットフォーム・アーキテクチャの相違 (ビッグエンディアン → リトルエンディアン)
- 制御言語JCL/CLの相違 (JCL/CL → Shellや制御アプリケーションプログラム)
- 文字コードの相違 (JEF/EUC/SJIS → Unicode)、マルチフォーマットの取り直し技術
- DBMSの相違 (索引ファイル/階層型/ネットワーク型 → リレーショナルDB)
- 帳票の相違 (ビジネスサーバの対話型帳票作成ツールは作り直し、富士通帳票定義体は流用可能)

Unicode選択の理由



■ 企業のグローバル化

■ 企業間の連携データがUnicodeの方向

■ 各業界もUnicodeを採用の方向

株券電子化に合わせて、証券保管振替機構ではUnicodeを採用

■ Vista、Windows Server 2008からJIS2004をサポート

JIS2004(第3水準、第4水準漢字)の一部はUnicodeの拡張面に配置

■ 政府がJIS2004を推進

電子政府、国語政策、人名漢字(戸籍)などで従来の文字コードに含まれない漢字を利用

■ 移植性の向上(プラットフォーム無依存)

3.1 Unicode化への留意点

- マルチフォーマット項目に対する注意
- システム運用での注意

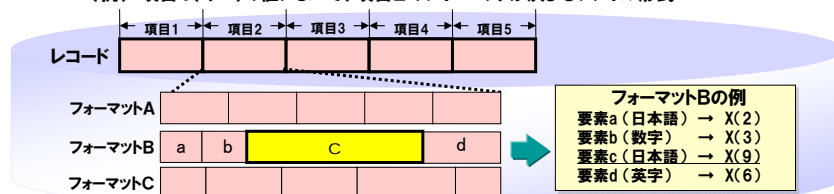
マルチフォーマット項目に対する注意【課題】FUJITSU

■ Unicodeはマルチバイトを特徴とし、全角文字でも固定長で扱うことができない。

➢ 項目を固定長(バイト長)で取り回すマルチフォーマット・ファイルが利用できない。

【マルチフォーマット形式のデータ項目とは】

(例) 項目1(キー)の値によって、項目2のフォーマットが決まるファイル形式



問題点

UnicodeではCOBOL X属性に、1~3バイトの文字が格納できるため固定長とならない

【具体例】

(例)要素cの格納情報

①	富	士	通
②	A	B	C
③	3	μ	g
	3bytes	2bytes	3bytes

可変

- 要素cが可変長となり、COBOLでの利用が不可能！
- 要素dの開始位置は固定ではない。

Unicodeの特徴

- 典型的なマルチバイト/文字コードを特徴とするコード体系
 - COBOLアプリの開発規約として、文字コードのバイト・ハンドリング規約を定め、プロジェクト全体で徹底しないと、AP品質の確保が困難になる

特徴	コード	JEF	Unicode		S_JIS	EUC (U90)
			UCS2	UTF-8		
1文字の大きさ(バイト数)	半角文字	1バイト/文字	2バイト/文字	1バイト/文字	1バイト/文字	1バイト/文字
	半角カナ	1バイト/文字	2バイト/文字	3バイト/文字	1バイト/文字	2バイト/文字
	全角文字	2バイト/文字	2バイト/文字	3バイト/文字	2バイト/文字	2バイト/文字
	ギリシャ文字	2バイト/文字	2バイト/文字	2バイト/文字	2バイト/文字	2バイト/文字
	演算子	2バイト/文字	2バイト/文字	2,3バイト/文字	2バイト/文字	2バイト/文字

文字コードの並びの相違

SORT
 2357ABCDEF

D7F5A2C3BE
 SORT
 ABCDEF2357

JEFコードと異なり、英字→数字の順となる。

「空白」に関する注意

- 他のコード系の経験・常識を無造作に当てはめるべからず
 - Unicode : 連続する半角空白2文字は、決して全角空白とはならない
 - 他Code : 連続する半角空白2文字を都合よく全角空白と解釈できる

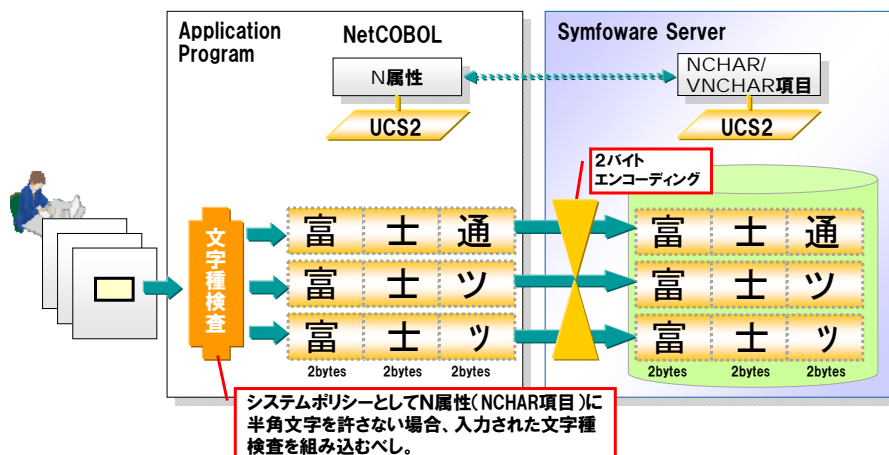
コード種	全角空白と半角空白の関係	空白文字コード (16進数)
Unicode	<div style="border: 1px solid gray; padding: 2px; display: inline-block;">全角空白</div> ≠ <div style="display: inline-block; vertical-align: middle;"> <div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-right: 5px;">半角空白</div> <div style="border: 1px solid gray; padding: 2px; display: inline-block;">半角空白</div> </div>	<div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-right: 5px;">30</div> <div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-right: 5px;">00</div> ≠ <div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-right: 5px;">20</div> <div style="border: 1px solid gray; padding: 2px; display: inline-block;">20</div>
JEF	<div style="border: 1px solid gray; padding: 2px; display: inline-block;">全角空白</div> = <div style="display: inline-block; vertical-align: middle;"> <div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-right: 5px;">半角空白</div> <div style="border: 1px solid gray; padding: 2px; display: inline-block;">半角空白</div> </div>	<div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-right: 5px;">40</div> <div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-right: 5px;">40</div> = <div style="border: 1px solid gray; padding: 2px; display: inline-block; margin-right: 5px;">40</div> <div style="border: 1px solid gray; padding: 2px; display: inline-block;">40</div>

COBOLアプリにて、表意定数SPACEを転記しているロジック箇所は要注意。

COBOL N属性に入る文字コードの注意



- COBOL N属性には、Unicodeの場合は全角文字だけでなく「半角文字」も入る
N属性(NCHAR項目)は、2バイト・エンコーディングの文字セットを格納できる項目属性
半角文字は、EUCでは1バイトの文字セットのためNCHAR項目には格納できない(エラー)



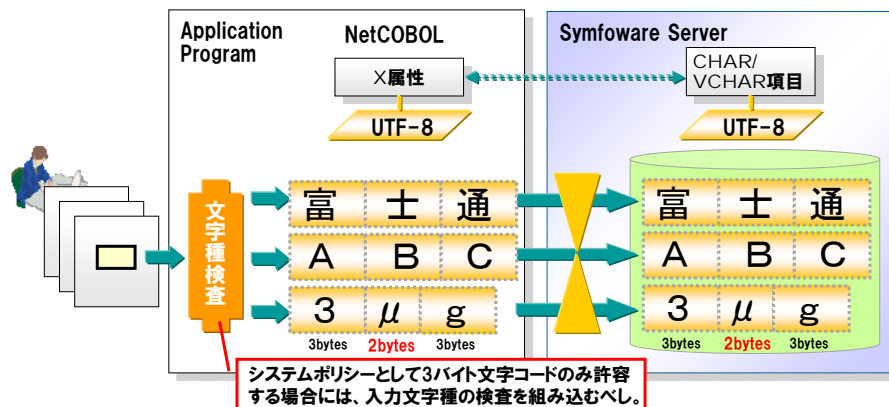
30

Copyright 2008 FUJITSU CIT LIMITED

COBOL X属性に入る文字コードの注意



- COBOL X属性には、1～3バイトの文字コードが混在して入る
 - 固定バイト長/文字コードを前提としたコーディングをするべからず
 - ASCII文字(いわゆる1バイト英数字)は、1バイト文字コード
 - 半角カナを含めた日本語文字は、3バイト文字コード
 - キリシヤ文字、キリル文字、一部の演算子含む記号(例: ÷, ±)は、2バイトコード



31

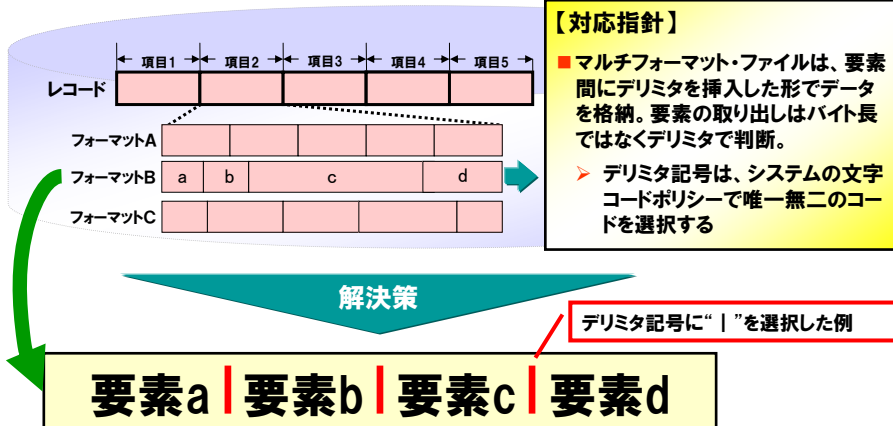
Copyright 2008 FUJITSU CIT LIMITED

マルチフォーマット項目に対する注意【施策】 FUJITSU

■ デリミタ挿入方式を採用しマルチフォーマット形式に対応

- 項目の判断は、バイト長ではなくデリミタで判断する

【マルチフォーマット利用におけるデリミタ挿入方式】



32

Copyright 2008 FUJITSU CIT LIMITED

FUJITSU

3.1 Unicode化への留意点

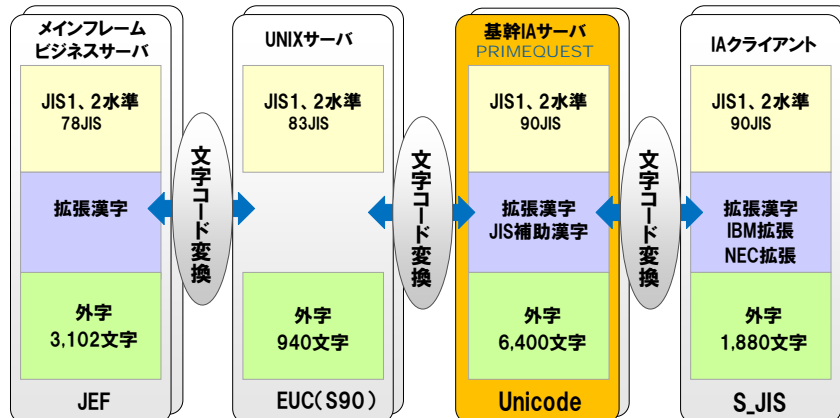
- マルチフォーマット項目に対する注意
- システム運用での注意

33

Copyright 2008 FUJITSU CIT LIMITED

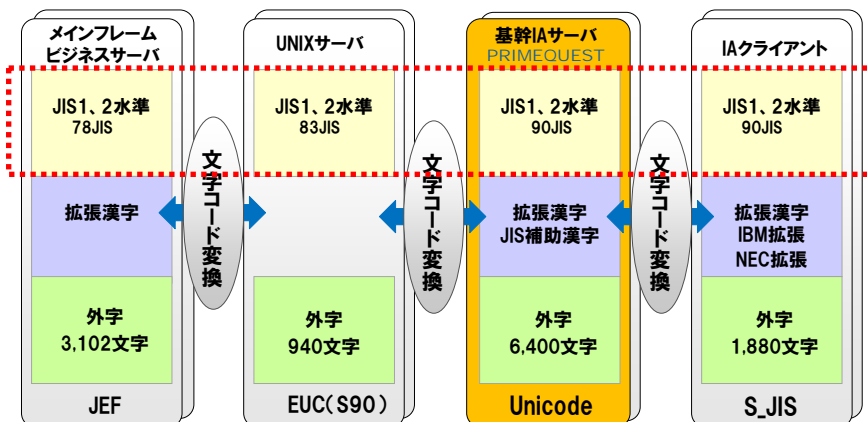
システム運用での注意【課題】

- Unicodeへ移行しても、プラットフォーム間で文字セットが異なると
 - 文字セットに含まれない文字を扱うと文字化けのトラブルに
 - 各システムのいろいろな場所で文字コード変換を行う必要がある



システム運用での注意【施策1】

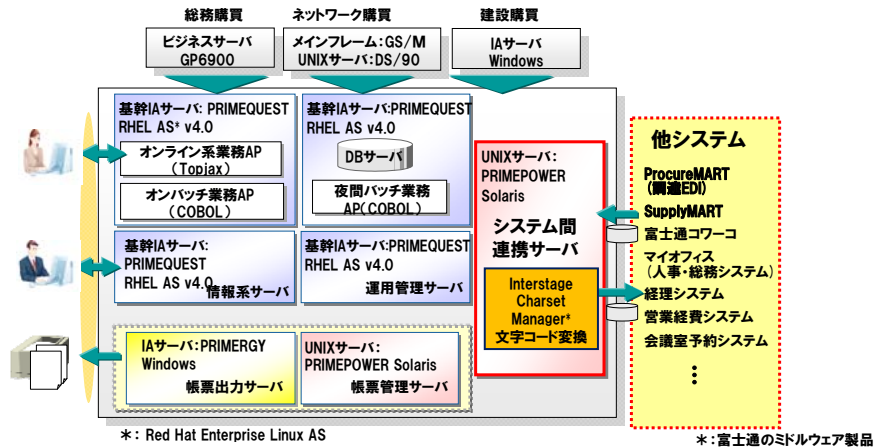
- 文字セットに含まれない文字を扱うと文字化けのトラブルに
 - ⇒ 文字セットが複数ある場合、システム全体で運用(受け渡し)する文字は、各セット共通の文字とする



システム運用での注意【施策2】



- 各システムのいろいろな場所で文字コード変換を行う必要がある
 - ⇒ システム間連携サーバを配置し、コード変換を集約
 - コード変換は Interstage Charset Manager*で実施



■商標について

- Microsoft、Windows、Windows Vistaは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- UNIX は、米国およびその他の国におけるオープン・グループの登録商標です。
- SolarisおよびすべてのSolarisに関連に関連する商標及びロゴは、米国及びその他の国における米国Sun Microsystems, Inc.の商標または登録商標です。
- Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です。
- Red Hat、RPM およびRed Hat をベースとしたすべての商標とロゴは、Red Hat, Inc.の米国およびその他の国における登録商標あるいは商標です。
- JavaおよびすべてのJava関連の商標およびロゴは、米国およびその他の国における米国Sun Microsystems, Inc.の商標または登録商標です。
- Oracleは、米国Oracle Corporationの登録商標です。
- Interstage は、富士通株式会社の登録商標です。
- 会社名、製品名等の固有名詞は各社の商号、商標または登録商標です。
- その他、本資料に記載されている会社名、システム名、製品名等には必ずしも商標表示 (TM・®) を付記していません。

The Fujitsu logo is centered within a black rectangular border. It features the word "FUJITSU" in a red, serif font. Above the letter "J" is a red infinity symbol.

THE POSSIBILITIES ARE INFINITE