COBOLの現状とこれから

2014-05-23

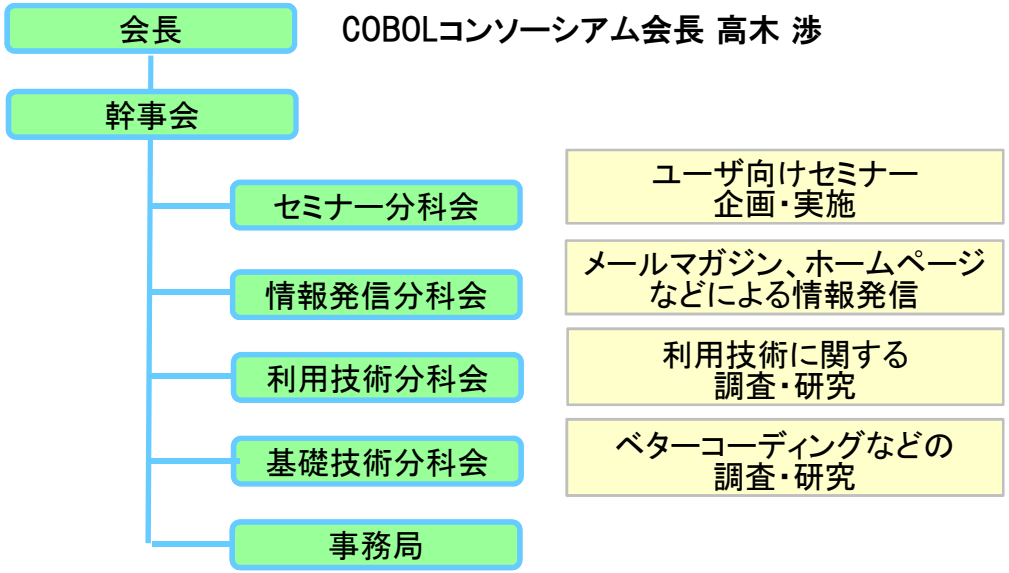
COBOLコンソーシアム会長　高木渉
ISO/IEC JTC 1/SC 22/WG 4 (国際COBOL委員会) 主査
情報処理学会 情報規格調査会 SC 22/COBOL WG 主査

（株）日立製作所
情報・通信システム社　ITプラットフォーム事業本部
開発統括本部　ソフトウェア開発本部　第2AP基盤ソフト設計部

COBOLコンソーシアムの紹介

COBOL言語を広く普及させ、COBOLユーザの利益を守るために設立された非営利団体（2000年設立）

COBOLコンソーシアム組織



COBOLコンソーシアム会員

- 東京システムハウス株式会社
- 日本電気株式会社
- 株式会社 日立製作所
- 富士通株式会社
- 株式会社システムズ

COBOLコンソーシアム準会員

- 日本ユニシス株式会社
- ユニアデックス株式会社
- 株式会社 ビーエスピー
- 株式会社日立ソリューションズ

COBOLコンソーシアムセミナー

[日時・場所]　2014年4月24日(木) 13:00-17:00　銀座フェニックスホール

[主催]　COBOLコンソーシアム/日経BP

[協賛]　NEC，東京システムハウス，日立，富士通，マイクロフォーカス

[参加者数]　219名 (満席)

[基調講演]　岩上 由高 氏（ノークリサーチ シニアアナリスト）

1. COBOLの誕生

1-1 COBOLの仕様は1年で作られた

◆1959年4月8日
- 共通業務用プログラム言語の必要性についての集まり

◆1959年6月23-24日
- Short-Range委員会初回会議

◆1960年1月7-8日
- Short-Range委員会の仕様をCODASYLが承認
  (CODASYL: COnference on DAta SYstems Languages)

◆1960年4月
- 仕様書発行 (米国Government Printing Office)

(参考: Jean E. Sammet, The Early History of COBOL, History of Programming Languages, ACM, 1978.)

1-2 COBOL仕様設計時の方針

- ◆業務用データ処理向け
- ◆言語が計算機から独立 (移植性がある)
  - ●移植性は1950年代から課題とされていた
- ◆エレガントさより，使いやすさ・読みやすさ
  - ●書くことより読むこと
- ◆簡単な英語で書く
  - ●経営側の人でも読める

1-3 COBOL-60の功績

- ◆データ部と手続き部を分離
- ◆データ構造を多段に階層化したレコード定義
- ◆PICTURE文字列によるデータの定義
  - ●当時参考にした言語から採用
- ◆変数名は30文字まで
- ◆数字項目は十進数の18桁まで
- ◆英語に近い表現で，さらに補助語を導入

  例: READ ファイル名 **RECORD** INTO 変数

2. 社会に浸透しているCOBOL

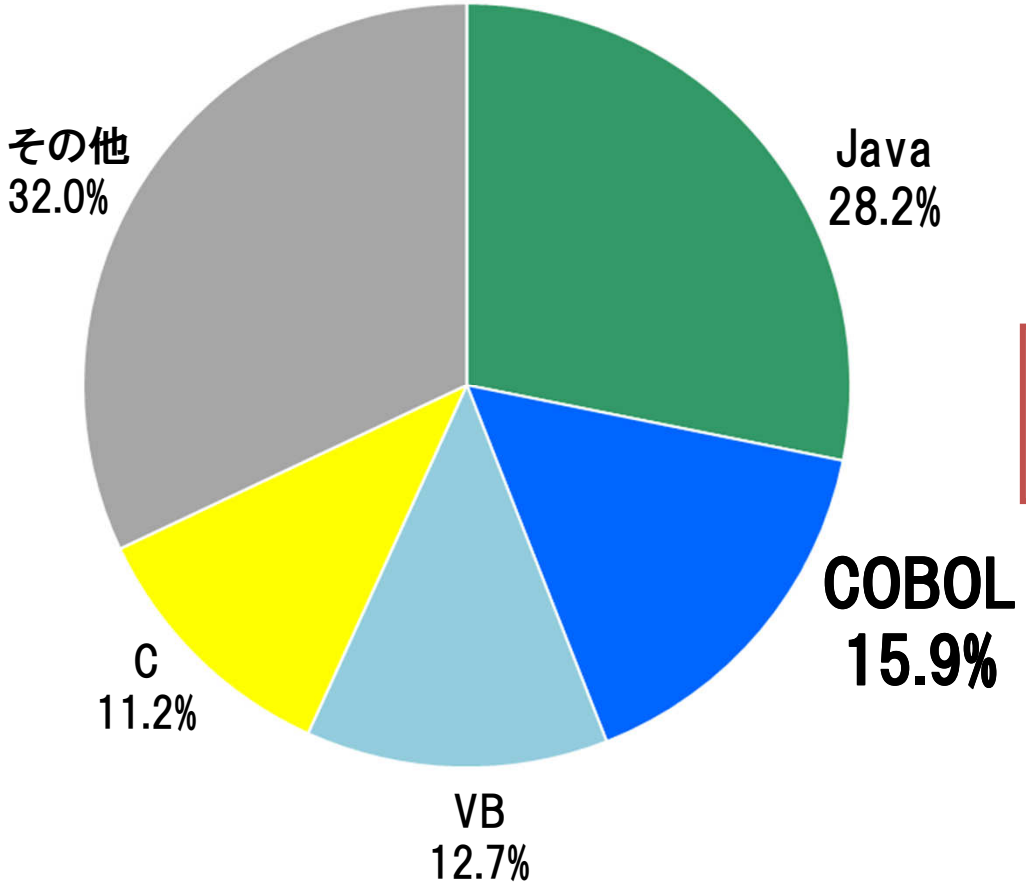
2-1 社会基盤を支えるCOBOL

◆大規模から中小規模まで，生活に浸透
- ●さまざま行政システム
- ●さまざま企業内のシステム
- ●銀行オンラインシステム
- ●座席予約システム

:

◆すでにインフラとして実際に稼動
- ●目立たないところで，役に立っている

2-2 開発に用いた言語（日本国内）



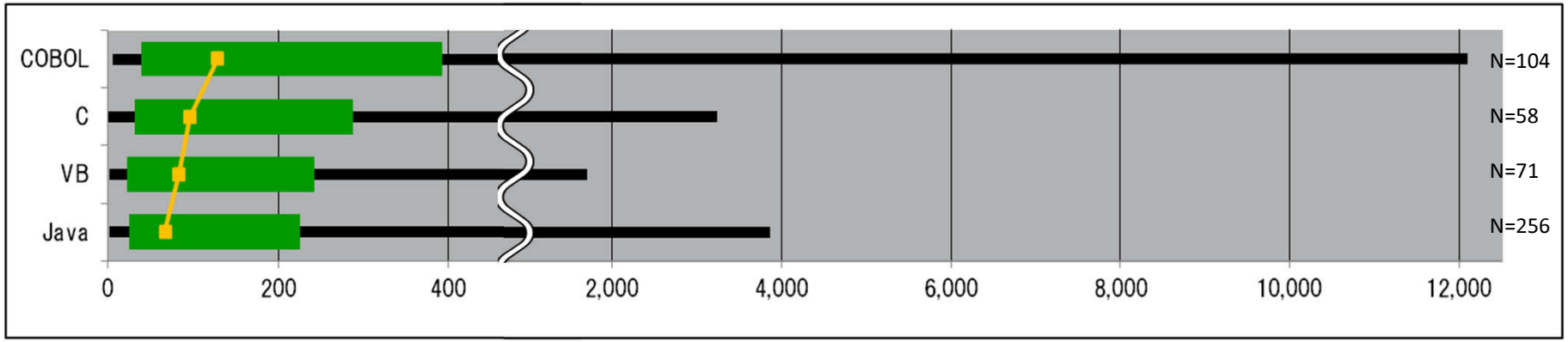
プロジェクトにおける採用言語
第1回答のみ集計
プロジェクト数: 2,918/3,089件

データ:「ソフトウェア開発データ白書 2012-2013
独立行政法人 情報処理推進機構(IPA) ソフトウェア・エンジニアリング・センター(SEC)

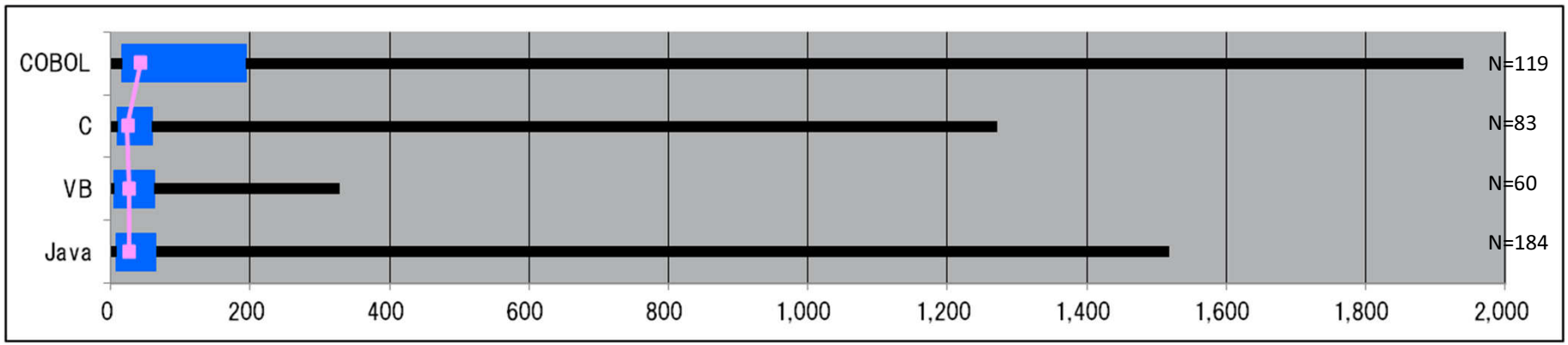
2-3 大規模システムに使われるCOBOL

新規開発（プロジェクト数：489件）

（単位：キロ ソースステップ数）



改良開発（プロジェクト数：446件）



箱ひげ図の箱は、全体の25%から75%に相当するデータ範囲。折れ線グラフは中央値。

データ：「ソフトウェア開発データ白書 2012-2013
独立行政法人 情報処理推進機構(IPA) ソフトウェア・エンジニアリング・センター(SEC)

2-4 米国調査によるCOBOLの利用状況

プログラム言語別の利用状況（複数回答）

Visual Basic	67%
COBOL	62%
Java	61%
JavaScript	55%
VB.Net	47%
C++	47%
Perl	30%
C	26%
C#	23%
ColdFusion	15%
PHP	13%
Fortran	7%
PL/I	5%
Python	5%
Pascal	4%
Ada	2%

＊資料：Computerworld米国版「Computerworld Survey」(2006年3・4月調査)

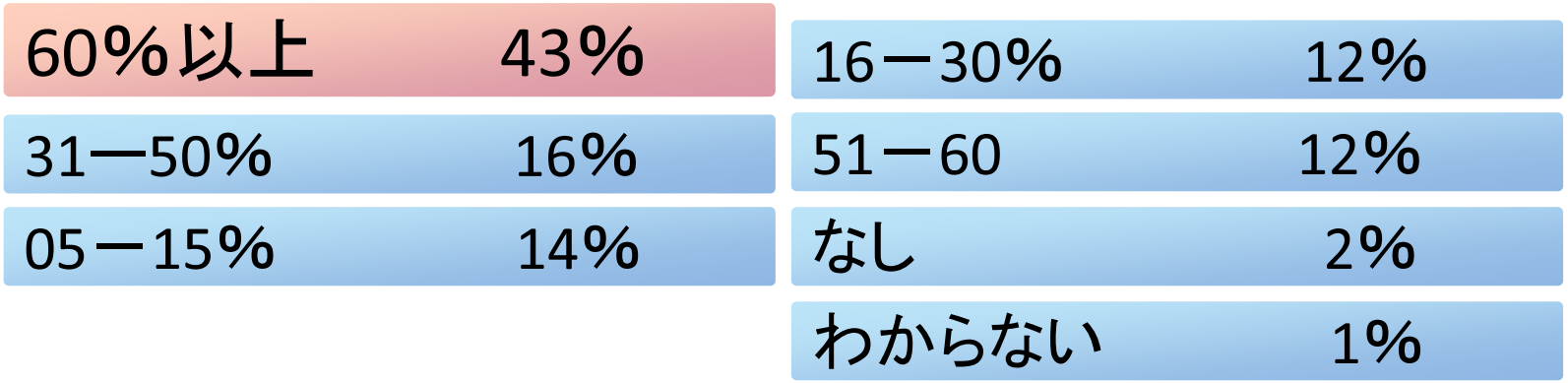
出典：月刊 ComputerWorld 2007 March　特別企画「2007年、COBOLは死せず」

2-5 米国調査によるCOBOLの利用状況

新しいアプリケーションの開発におけるCOBOLの利用状況

＊資料：Computerworld米国版「Computerworld Survey」(2006年3・4月調査)

社内のアプリケーションにおけるCOBOLアプリケーションの割合



＊資料：Computerworld米国版「Computerworld Survey」(2006年3・4月調査)

出典：月刊 ComputerWorld 2007 March　特別企画「2007年、COBOLは死せず」

2-6 世界で稼働するCOBOLの量

COBOLは世界のほとんどの
商用基幹アプリケーションを稼動

- **ビジネス・データの75%**
- **金融取引では90%を処理**
- **2,000億行のコードと推定**

出典：COBOL誕生50周年記念セミナー～社会を支える“ことば”。これまでも、そしてこれからも～（2010年4月16日配布資料）
日本アイ・ビー・エム「COBOL開発ライフサイクルを刷新するエンタープライズ・モダナイゼーション」

3. COBOLの良さ

3-1 COBOLの特徴 (1/2)

◆レコード入出力指向

- 入力レコードを駆動源とした処理を書きやすい

◆型の概念の絶妙な弱さ

- メモリ範囲にレコード定義を重ねるだけで各フィールドが使えるようになる
- フィールド間で型が異なれば，MOVE文はデータを型変換する

3-2 COBOLの特徴 (2/2)

- ◆基本的に演算は十進数で
  - ●計算機は二進数でも，人は十進数で生活する
- ◆静的言語
  - ●アプリケーションが稼働時に使うメモリ量を想定可能
  - ●読解時に注目すべきステートが少ない
- ◆一つの文に多くの処理を詰め込めない
  - ●読む速度で理解する
- ◆言語仕様の互換性が長期間保たれる

3-3 COBOLは長期稼動に有利

◆長い目で測る生産性
- ●初期開発の時
- ●変更を繰返したソースの更なる変更の時

◆プログラムの変更に必要な要素
- ●変更対象のプログラムの振舞いを理解する
- ●変更方法を考案する
- ●変更が正しいことを確認する

(変更箇所の正しさ + 非変更箇所の振舞いの保存)

4. 国際規格の動向

4-1 関連する標準化団体

- ◆ISO/IEC JTC 1/SC 22/WG 4
  - ●国際標準化団体
  - ●大きな方針を決定
  - ●日本，米国，英国，オランダ，カナダ
- ◆ISO/IEC JTC 1/SC 22/WG 4/OWG-1
  - ●実際の規格文面を作成
- ◆米国INCITS PL22.4 (OWG-1と実質同一)
  - ●規格の誤り指摘に対する検討

4-2 次期規格について

- ◆日程
  - ●2014年6月 国際規格発行予定
- ◆2002年規格の仕様のオプショナル化あり
  - ●画面入出力機能，ロケールなど
- ◆主な追加機能
  - ●IEEE754 の十進/二進浮動小数点数の扱い
  - ●DYNAMIC LENGTH基本項目
  - ●容量可変表 (既存の可変長項目より柔軟)

4-3 規格委員会の様子

オブジェクト指向言語仕様の議論から

◆英語らしさを気にする

- `CLASS-ID  A INHERITS FROM B.`
  - ➢補助語のFROMを付ける/付けないで意味が異なる。
- `INVOKE obj-ref-1 ”method-1” USING P1 P2 P3 RETURNING R1`
  - ➢語順が INVOKE ”method-1” on obj-ref-1 でないと英語でない。
- `obj-ref-1 :: ”method-1” (P1 P2 P3)`
  - ➢ :: などという目がちかちかする記号は導入したくない。

5. これからのCOBOL

5-1 COBOLの再評価

- ◆データの持ち方は実は効率がいい
  - COBOLのレコードデータは，入出力効率がいい。フィールドを分離する記号は処理オーバーヘッドになる。
- ◆COBOLプログラマーはすぐに養成できる
  - 他言語プログラマーがCOBOLを習得するのは難しくない。ハードルはモチベーションの持ち方。
  - COBOL “も” 使える技術者を育てよう。
- ◆既存プログラムの読みやすさは保守に有利
  - 保守性を向上するテクニックがあっても，プログラムを理解するのに苦労していては効率が上がらない。

5-2 今後のCOBOLについて (私見)

- ◆COBOLの言語仕様自体に，あまり大きな拡張はないだろう。
- ◆新技術は，（残念ながら）まずCOBOL以外の言語で利用可能になることが多いだろう。
- ◆COBOLと他のシステム/技術との連携可能性が，COBOLを使い続ける理由になるだろう。
- ◆COBOLのシステムは，社会基盤として今後も我々の生活を支えるだろう。

• OracleとJavaは，Oracle Corporation 及びその子会社，関連会社の米国及びその他の国における登録商標です。
•その他記載の会社名，製品名は，それぞれの会社の商標もしくは登録商標です。

END