

レガシーシステムの見直し

わが国は、現在においてもメインフレームの使用率が高く（注1）、これらの運用・保守を行っている団塊世代のエンジニアが、2007 年以降に大量にリタイアすることから、2007 年問題と言われている。一方、メインフレームは今後もなくならないという見方もあり、メインフレーム = レガシーシステムとは言い切れない。

そこで、メインフレームの中で用いられているコンピュータ言語に注目し、レガシーシステムの定義づけを試みる。

（注1）「ダウンサイジング」という言葉が登場して 10 年以上立つが、日本では今でもメインフレームが全サーバ市場における売上の 30% 以上を占めている。その他地域が 10 ~ 15% 程度となっているのに対して、2 倍以上の数値である。

（http://www.atmarkit.co.jp/fsys/it_market/it_market014/it_market014-01.html）

米国や欧州では、メインフレームを販売する企業は 1-2 社程度に絞られたが、日本では、富士通、日本 IBM、日立製作所、日本電気、日本ユニシスの 5 社がメインフレーム事業により利益を上げており、Unix サーバや PC サーバも並行して販売しているが、完全な移行は推進していない。

1 COBOL 中心の業務システム形成

(1) コンピュータの導入期

1960 年代に始まる業務へのコンピュータ導入に際しては、次のような変遷があった。

- ・ アセンブラ
- ・ COBOL（ビジネス中心）
- ・ FORTRAN（科学技術計算中心）
- ・ PL/1（金融（IBM 系）システム中心）

この中でも特に、COBOL は、Common Business Language という名が示すとおり、業務システムのデータ処理に適した言語であったため、多くのコンピュータシステムに適用されてきた。

この COBOL 言語で育った企業内の情報システムエンジニア（COBOL 言語世代）は、主に昭和 20 年代前半生まれの団塊世代（現在 50 歳後半）と一致している。

(2) 現在のメインフレームの言語

その後、いくつかの言語（C、Visual Basic、Java など）が開発されてきたものの、COBOL 中心の運用や再構築の構造は、現在においても基本的に変わっていない。その理由としては、

コンピュータシステムの再構築には、業務系の知識を必要とするが、その社内の中心スタッフは COBOL 言語世代である。

企業における IT ガバナンスの必要性が出てきたのは、1990 年代後半からであり、それまでは、メインフレームからオープンシステムへの移行に伴うリスクやコストが高く、企業メリットがなかった。

などであり、その結果、メインフレームにおいては、現在においても COBOL が中心的な言語となっている。

2 レガシーシステム = COBOL 運用のメインフレーム

(1) COBOL 世代のリタイア = 2007 年問題

しかし、これらの COBOL 言語世代が、あと 2~3 年で現役を退き始める。これがいわゆる 2007 年問題であり、わが国における COBOL 運用のメインフレームをメンテナンスできる人材が急速に減少していく可能性がある。

(2) 世界の動向

米国などでは、わが国より早くメインフレーム中心から脱却していることに加えて、人材層も厚い。すなわち、米国人システムエンジニアに加えて、インド系などの海外からのシステムエンジニアを受入れている。

しかし、その米国においてすら、数年前から COBOL 言語世代のリタイアに対する企業リスクに注目が集まっている。(Gartner Research 2002 9月27日等)

(3) わが国におけるレガシーシステム

米国ほど人材の国際化やオフショア(海外企業へのアウトソーシング)が進まず、しかも、メインフレームの利用比率が高いわが国においては、この COBOL 言語世代のリタイアによって、個々の企業だけでなく、社会全体が大きなリスクに直面する可能性がある。

レガシーシステム対応は、legacies Migration と呼ばれており、リサーチ会社やベンダーなどから、いくつかの方向性が提案されている。

<http://japan.cnet.com/news/ent/story/0,2000047623,20063445,00.htm>

<http://www.keieiken.co.jp/monthly/repo0310/03101-3.html>

<http://japan.cnet.com/news/ent/story/0%2C2000047623%2C20072203%2C00.htm?ref=rss>

<http://japan.cnet.com/news/biz/story/0%2C2000050156%2C20064954%2C00.htm?ref=rss>

(補論：広義のレガシーシステムとしての論点)

「メインフレームとCOBOL」に限定した場合の進行上の問題点

2007年問題や、オブジェクト指向的設計が困難な事から、「新しいニーズへの対応が困難なひとつの切り口」ではある。しかし、ダウンサイジングされた現在主流となっているオープン系システムの大半でも、他方で論議されている企業構造のモジュール化やSOA化への対応など、ユビキタス時代の新たなニーズには対応が難しい。

広義なレガシーシステムの定義

「逐次的処理の必要なクローズなシステム」

前回議論の文脈と「ダイナミックなコラボレーションの促進」を念頭に置くと、「モジュール化されたオープンなシステム」との対比が求められる。

主要な判別点は「システム間連携が可能か(容易か)どうか」となる。

「COBOL運用型メインフレーム」に関する現状と課題

(1)SOA化への対応力

メインフレームやCOBOLに対しても、既存の情報資産を活かしつつWebサービスやSOAに対応させるソリューションも出始めており、一概にSOAへの対応が困難であるとは言えないが、業務の設計段階でのモジュール化の検討が必要であり、既存のシステムをそのまま維持してSOAに対応できる企業は少ないと思われる。

(2)「2007年問題」

2007年問題は、情報システム分野以外でも「引退するベテランの暗黙知をどのように継承するか」について、2010年問題などとして議論されている。

しかし、基幹系システムにおける主要な問題は「ベテランの不在」よりも「システムの肥大化」にあるといえる。構築当初はシステムの規模が小さかったため、全体像が見えていたが、継ぎ足しによる機能追加を個別部門で実施していったために、(ベテランでも)全体像を把握できなくなってしまうと考えられる。