うこだより

第７号

編集発行人
清水吉男

(株)システム クリエイツ
横浜市緑区中山町 869-9
電話 045-933-0379
FAX 045-931-9202

システム設計講座

今回は趣向を変えて『ＣＡＳＥ』について触れて見ます。ＣＡＳＥの歴史は一九八〇年代初期に文書や幾つかの分析手法に基づくダイヤグラム作成ツールとして始まり、パソコンをベースにしたソフトウエア開発ツールで、分析設計の自動化ツールとして誕生したものです。そして『ＣＡＳＥ』という名称がつけられたのはほんの数年前です。

ＣＡＳＥとは **Computer Aided Software Engineering** の頭文字を採ったものです。ソフトウェアの歴史は、システムの設計からプログラムの作成そしてテストという作業を効率よく行うための工夫の歴史でもあります。

多くの人がこの「ソフトウェア・エンジニアリング」について考え、提案してきたわけですが、ＣＡＳＥはこれらの一連の作業をコンピュータを使って自動化・効率化するための方法で、このために作られた開発支援システムを総称してＣＡＳＥと呼んでいます。まさに高性能なコンピュータとそのネットワークそして一人一台の端末割り当てと言う時代を背負って出てきた手法といえるでしょう。

§ § § § §

現在一部で使われている、フローチャートからプログラムソースを自動生成したり、データの定義書からテストデータを生成するいわゆるツールキットの他に、ワークベンチと呼ばれる「ＣＡＳＥツール」は、

一、分析を行うためのツール。

二、分析に基づいて設計仕様書（プログラムに関する仕様書、データに関する仕様書その他）を作成するためのツール。

三、作成されたプログラム仕様書からプログラムソースを生成するためのツール。

四、データ仕様書から有効なテストデータを生成するためのツール。

五、テストを効果的に行うためのツール。

等で構成され、各々が別々のものではなく、『レポジトリ』というデータベースを間に置いて互いに関連しあっています。その結果Ａ氏の過去の成果を簡単にＢ氏が活用することができる様になっています。

§ § § § §

この様に、ワークベンチ型ＣＡＳＥツールでは、全て一、及び二、で作られた仕様書に基づいており、この仕様書の出来具合が結果を左右することになります。勿論、データの構造が一致していないといった所謂初歩的ミスに関しては、これらのツールによって事前に発見されるでしょうが、設計そのものの精度は必ずしもカバーされません。いくらコンピュータが作ってくれるといっても、どの様なものを作りたいのかを正確に表現できなければ無理な話です。まだまだＳＦ映画の様に設計者の考えたこと以上のものが作り出される状況ではありません。

§ § § § §

ＣＡＳＥがいくら素晴しい道具でも、それを使う人にＣＡＳＥツールの機能を十分に発揮できるだけの能力と経験がなければ役に立ちません。すなわち、要求仕様や現実の問題を分析して設計仕様書を導く能力がなければ、如意棒もただの棒切れに過ぎないのです。現実に現在出回っているワークベンチ型のＣＡＳＥツールは、幾つかの分析設計手法に基づいており、その手法に習熟していなければ望みの結果を得ることは難しいでしょう。しかもそれらの手法はそれぞれ得意分野を持っており、必ずしも全ての分野に適応できるものではありません。そのために当人が抱えている問題を解くにはどの手法が適しているのかを判断する必要があると同時にＣＡＳＥツールも選ばなければなりません。とは言ってもそれらの手法は互いにかけ離れた位置にあるのではなく、例えば処理手続きに重点を置くかの違いであり、一つを周知していれば他の手法をマスターすることはそれ程障害はないと考えられます。

§ § § § §

アメリカに於ては高度の分析設計能力をもち、ソフトウェアの生産性向上と品質改善に強い意欲を持ったシステムアナリスト達が、既にワークベンチ型のＣＡＳＥツールを使って成果を上げているようです。しかもプログラマーを一人も使わないで全てＣＡＳＥツールでプログラムを自動生成している例もあります。当然のことに彼等は設計だけではなくプログラムを書くこともできます。設計したものがどの様にプログラムに変化していくのか、設計段階でどうしても表現出来なかった部分は、プログラムになったときにどこに位置しているのかが分からなければならず、時としてプログラムを自ら手直しする必要があるからです。

§ § § § §

ただ、現在作られているＣＡＳＥツールは殆どは欧米で作られており、このままでは仕様書における言語の問題が障害になるようです。日本で扱える有効なＣＡＳＥツールの開発が待たれるところですが、パソコンやＷＳがここまで普及した状況では、この考え方だけでも取り入れていくことは可能でしょうし、そうすることによって順次新しいソフトウェア・エンジニアリングの技法の恩恵を受けることができるでしょう。◆

年少者人口減少の中で

▼昨年（平成元年）一〇月一日現在の日本の人口ピラミッド（推計）をみると、その年の出生者数は第二次ベビーブーム世代の約六割に止まっている。しかも通常なら四、五年前から少し回復しても良いものが昭和五十年以降減少し続けている。予想以上に若年労働者の現象は深刻である。

▼これを受けて昨今ソフト業界も若年労働者の確保に躍起になっている。入社式がディスコパーティとなり社長自らレースカーで会場に乗り込むなど、注目を引かんとすること涙ぐましい努力である。背に腹は代えられないのであろう。

▼参入障壁が低ければレベルが下がり、レベルが下がれば付加価値も低下していくのは当然の理である。１００点の人がいなくても８０点の人が数人集まれば同じ様な結果を生み出すことは出来るかもしれないが、３０点の人が何人集まっても代わりは勤まらない。

▼若年労働者人口が減少していく中で、新しい人も既にソフトの制作に携わっている人も、一人一人のレベルを上げ、付加価値の高い仕事をくたびれることなく長く続けてもらうことが本筋であり、そのために今何を為すべきかが問われているのではないか。