

# 生産スケジューラー Asprova APS の設計内容とデモンストレーション

\*高橋邦芳 (アスプローバ株)

## 1. 開発の起源

Asprova の開発の起源は 1990 年前半の人工知能ブームまでさかのぼる。当時、筆者は Lisp を用いたエキスパートシステム構築ツールの開発をしていた。しかし、1993 年ごろ人工知能ブームは去り、次の開発テーマを検討せざるをえなくなった。エキスパートシステムには診断型、設計型、計画型などの分類があり、計画型エキスパートシステムのひとつとして生産スケジューリングがあった。日経 AI や新聞など産業界の情報を元にマーケティング調査を実施し、マーケットの要望が高く、かつ、ユーザにとって金銭的な効果大きく、かつ、実現可能なものとして、生産スケジューラーを開発分野として選定した。

## 2. 開発の歴史

最初のバージョンは 1992 年頃、MS-DOS 上の Lotus 1-2-3 を用いた生産計画表の作成プログラムである。これは Lisp を用いて各種ケースを枝狩りしながら解を出すものであった。これには、2 つの問題点があった。ひとつは、対応する製品数に上限がある点。もうひとつは、処理速度が遅い点である。このため製品数に限界がなく、しかも、超高速で動作することを設計目標とした。Lisp をやめて Borland C を用いて、ガントチャートが表示できるバージョンを開発した。これにも、ひとつの問題点があった。それは、メモリー空間が不足するという点である。次にメモリー空間を確保するため Sun Spark Station の X Window 上に移植した。しかし、これにもひとつの問題点があった。既に Windows 3.1 が出荷されていたため、ユーザはパソコン上で動くことを望んでいた。次に Windows 3.1 上で 32 ビットアプリケーションが開発できる WATCOM C386 に移植した。これで、ユーザに受け入れられるための基本的問題点は解消できた。

1994 年 Windows NT 上で Visual C++ を用いて作り直しを行った。このバージョンは爆発的にマーケットに受け入れられ 2001 年までに全世界で 800 サイト以上に導入された。2001 年それまでの反省点をすべて考慮して、ゼロから根本的に設計し直し、現在の Asprova APS を開発した。2008 年 2 月末日時点、通算で全世界 1200 サイトの導入実績に至っている。以下、Asprova APS の設計内容について解説する。

## 3. データ構造

### 3.1 オブジェクト、プロパティ

Asprova のデータは、全てオンメモリー上の多数

のオブジェクト内にある。この多数のオブジェクトは階層状にリンクされていると同時に、必要に応じて任意のネットワーク状にリンクされている。オブジェクト内には複数のプロパティが存在し、値はプロパティに格納される。プロパティの型の種類は内部的には 30 種類以上あるが主なものを下表にあげる。

Int4	整数
Time	日時
TimeSpan	時間
Enum	列挙値
Float8	実数
Symbol	文字列
Expression	式
SortExpression	ソート
GroupingExpression	グルーピング
Object	オブジェクト

表 1 プロパティの型

Expression 型は Excel のセルに記述する数式のような式を設定し、値は実行時に計算される。Object 型は他のオブジェクトへのリンクをつくり、オブジェクトの階層構造や任意のネットワークをつくる。オブジェクトの階層構造の例を以下に示す。

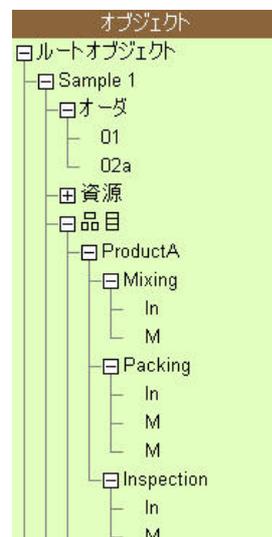


図 1 オブジェクトの階層構造

全てのオブジェクトは、ルートオブジェクトの下に存在する。Sample1 というプロジェクトの下にオーダー、資源、品目などのオブジェクトが存在し、その下に各データを格納するオブジェクトが存在する。品目の下には BOM 情報が格納される。

Asprova APS Ver5.3 では、オブジェクトの種類(クラス)は 158、プロパティの種類は 2446 である。

### 3.2 ユーザ定義クラス、プロパティ

クラスやプロパティはユーザが独自に追加可能で、追加するとテーブル表示や外部システムインターフェース時に、他のクラスやプロパティと同様に取り扱うことができる。

### 3.3 仮想プロパティ

ユーザが追加したプロパティが、他のプロパティの値から計算される場合は、その計算式を仮想プロパティに登録すれば、動的に計算された計算結果の値が格納されているかのように動作する。

## 4. 超高速な有限能力スケジューリングロジック

通常の生産スケジューラーの有限能力スケジューリングではデータ量(作業数)が多くなるに従い、リスケジュールの計算時間は指数関数的に長くなる。Asprova は独自の特殊ロジックにより作業数に対して計算時間はリニアである。1つの作業を割付ける時間は全体の作業数に関係なく一定である。(この特殊ロジックの内容は非公開である)

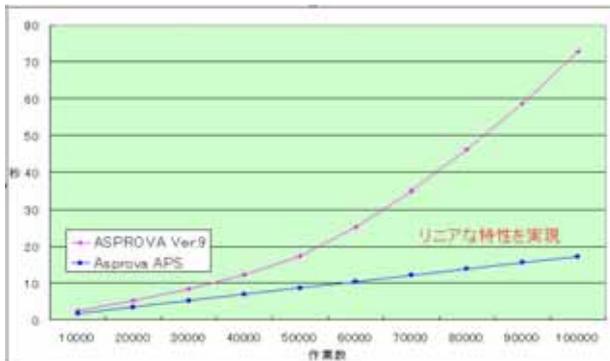


図2 有限能力スケジューリング処理時間

1つの作業を割付けるときには、割付先の資源の候補が複数発生する。特に、機械・金型・人員を同時に使う場合は候補数が非常に多くなる。例えば、機械・金型・人員とも100種類の候補がある場合は、全体の候補数は1,000,000となる。このような時にあらかじめ候補を発生する方式だと、多くのメモリーが必要で、かつ候補の発生に処理時間がかかる。Asprova では候補の発生を順次行い、途中で打ち切ることができる。このため、高速、高メモリー効率のロジックを実現している。

工程間を10分以上空けてはいけないなど、2つ以上の工程間の時間関係に制約があるとき、候補の発生はこの複数の工程で同時に行う必要がある。先に一方の工程の割付を決定してから、他方の工程の割付を決定する方式だと、最初の工程の割付の間違いが取り返しがつかず、よりより組み合わせができな。Asprova では2つ以上の工程の資源の候補を同時に発生させることにより、2つ以上の作業の割付資源の決定を最良化を可能にしている。

## 5. GUI

### 5.1 EXCEL ライクなテーブルの操作性

生産管理担当者は世界的に見ても Microsoft Excel を活用していることが多い。そのため、Asprova のテーブル(スプレッドシート)表示では、Excel と同様な表示、入力機能を備えることにより、ユーザが感覚的に簡単に使いこなせるように工夫している。

### 5.2 テーブル表示が合体したタイムチャート

スケジューリングをした結果を文字ベースの表示で見えていてもスケジュールの内容はユーザにはわかりにくい。高度な GUI を備えていることでユーザのスケジュール結果の理解を助ける。Asprova は、Microsoft Project と同様にチャート表示とテーブル表示が合体した GUI を採用している。GUI を見て直感的にスケジュール結果の良否を判断できることもよくある。



図3 テーブルとタイムチャート

## 6. 外部インターフェース

### 6.1 外部システムインターフェース

生産スケジューラーは、会社のERPシステムなどの既存システムとデータ連携をとって運用されることが多い。Asprova では、どのような外部システムとも連携が容易にできるように「フィールドマッピング機能」を備えている。



図4 フィールドマッピング

## 6.2 COM インターフェース

Asprova はカスタマイズしないで活用することを基本としている。しかし、スケジューリングロジックやユーザインターフェースなど、適用する工場の特殊な要望に応えるためにカスタマイズを行うことがある。前述した「式」の応用で解決するものは「式」を用いればよいが、「式」でも解決できない場合は、COM インターフェースを用いてプラグインを開発することができる。

Asprova のオブジェクト、プロパティはほぼそのまま COM インターフェースを通して公開されている。ユーザはこれらのクラスとプロパティを通して、Visual Basic, C++, C#などを用いてプラグインを開発する。

## 7. まとめ

弊社は設立以来 14 年以上、生産スケジューラーに特化した研究開発を続けてきた。1200 を超えるユーザサイトの要望を受け毎年バージョンアップしている。2000 種類以上あるプロパティには、この 1200 を超えるユーザの要望がノウハウ化して凝縮されているとあってよい。この 2000 種類以上のプロパティの開発の道のりは、14 年間にわたりユーザの要望を真摯に聞きながらの地道な開発の積み重ねである。

これまでの Asprova の利用方法は既存システム (ERP) とデータ連携しながら、スタンドアロンなシステムとして活用されてきた。将来の開発の方向性は、ネットワーク対応や Web 対応である。インターネットを活用することによって生産スケジューラーがどのように新しい役割を果たすか、まだ、検討を開始したところであるが、今後の研究開発の進展が楽しみである。

## 参考文献

- [1]. 今岡善次郎, "時間をキャッシュに変えるものづくり", 日刊工業新聞社