

XMLによる生産スケジューリング 標準インタフェースの開発

A Standard Interface with XML technologies for Production Scheduling Software

西岡 靖之
Yasuyuki NISHIOKA

法政大学工学部経営工学科 (〒184-8584 東京都小金井市梶野町 3-7-2
E-mail: nishioka@k.hosei.ac.jp)

ABSTRACT. Production scheduling software is a state-of-the-art technology in production management problems. Although some schedulers can successfully deal with the Japanese shop floor problems, they perform almost independently because of their poor communication standardization. Thus, in order to do more optimized production, we develop a standard interface for the production scheduling software. The interface is on the XML based technologies, so that various applications on Internet can be connected each other. In this project, several Japanese scheduling software vendors attend and implement the interface. As a result, we found out that the interface has huge potential to change business models of manufacturing enterprises.

1. 背景

製造業は変化の激しい市場ニーズに瞬時に対応するため、現場レベルのIT化による根本的な改革に真剣に取り組んでいる。本研究では、製造業の生産スケジューリングに関する業務をネットワーク上でバーチャルに統合した新しいビジネスモデルを実現するために、生産計画・スケジューリング問題のXMLによるインタフェース仕様の標準化を行い、それに併せたインタフェースモジュールの開発を行った。

生産スケジューラは今後のIT化された生産管理の中心的な位置付けを占めると予想されており、本研究の成果として開発されたモジュールと、それを用いたしくみが、ひとつの儲かるビジネスモデルとなる。ここで提案するしくみは、企業間の計画連携をとおしてさらに効果が高まるために、さらに多くの企業が参加し、自己増殖的にネットワークが発展していく可能性がある[1]。

現在、生産スケジューリング問題を対象とした標準化の動向としては、国内で唯一の標準化団体であるPSLXコンソーシアムが積極的に活動している。PSLXコンソーシアムは、2001年7月に、筆者らが中心となって設立した団体であり、現在ボードメンバーとして、国内の代表的なスケジューラベンダーをはじめ、SI企業やコンサルティング企業、そして大学などの研究機関など40団体以上が参加している[2]。

PSLXコンソーシアムは、基本的にボランティア団体であり、共通仕様を策定することを主な目的としているが、実用的な仕様を策定するためには、プログラムの実装を通してより制度の高い要求事項の洗い出しが必要となる。また、仕様が完成した以降、その仕様を世の中に普及させるためには、実際に動くモジュールの提供がなくてはならない。

以上のような背景から、本研究は、PSLXコンソーシ

アムでの仕様策定を先取りするかたちで、国内でもっともシェアの高いスケジューラベンダー4社の協力を得ながら、共通インタフェースモジュールの開発を行うことにした。そして、ここでの開発を踏まえて、各スケジューラベンダーの次期製品としてインタフェースを組み込んでもらうよう働きかける。

2. 目的

そもそも異なる生産スケジューラ間、あるいは生産スケジューラとERPやMESなど周辺アプリケーション間でコミュニケーションするためには、機能を極端に限定するか、個々のシステム間で個別にカスタマイズ等の開発作業が必要となる。したがって、できるだけ機能を限定せずに、かつ開発工数を減らすような標準インタフェースを開発することが本研究の目的となる。

異なるアプリケーション間でデータ連携を目指す試みとしてはEAI (Enterprise Application Integration)があり、RDB間でのデータ変換アダプターや、CORBAやDCOMを用いたRPCによるアプリケーション間の直接連携、そしてインターネット上にあるアプリケーション間の粗結合を目指したXMLおよびSOAPを用いた方式などがある。本研究では、基本的に、最後にあげたXMLをベースとする方式をとっている。

ただし、各アプリケーションがXMLをベースとしたインタフェースを個々に実装することは、開発者の負荷が非常に高くなるため、本研究では、アプリケーションの開発者に負担をかけずに、XMLをベースとしたアプリケーション連携の機能を組み込むことができるようなモジュールを開発することをメインの開発ターゲットとした。

本研究は、ソフトウェア単体としてはそれほど創造性はないが、開発するソフトウェアを要素としたビジネスモデル、あるいはWin-Winコラボレーションモデルの構

築が非常に独創的であり、成功すれば、わが国としては極めてユニークな事例となるだろう。そして同時に、このしくみは、スケジューラベンダーを核に、MESベンダ、ERPベンダ、SCPベンダなど、さまざまな企業をも巻き込んだ“和製ERP”バーチャルエンタープライズとして、現在まったくやられっぱなしの感がある欧米ITベンダーへの対抗勢力となることも十分に期待できる。

3. 開発内容

(1) 標準仕様の概要

PSLX コンソーシアムが現在公開している仕様としては、PSLX-01(グラントデザイン)、PSLX-02(オブジェクトモデル)、PSLX-03(XML 標準規約)、そしてPSLX-00(統一用語辞書)がある。本研究では、この中でPSLX-03(XML 標準規約)の部分を中心に対象とした[3]。

PSLX-03 で定義されている XML のタグは以下のような構成となっている。これらのタグを組み合わせることによって、生産スケジューリングに関するあらゆる情報を表現できるようになっている。これらのタグを用いて記述された XML 情報をアプリケーション間で交換する。本研究では、これらのタグセットを用いて、アプリケーション側が容易にデータ連携あるいはプロセス連携可能なインタフェースモジュールを開発した。

表 1 PSLX のタグセット

種別	例
制御用タグ	pslx,problem,protocol,method
基本データ	qty,char,time,duration
補助データ	unit,translate,scale,base,calc,arg
基本制約	min,max,earliest,latest,shortest,longest,candidate
基本要素	event,location,customer,supplier,item,resource,order,operation
属性要素	spec,mode,position,priority,penalty,description,display,color
関係要素	relation,parent,distance,condition,action
拡張属性要素	rate,price,lotsize,start,end,release,duetime,progress,load,stock,calendar
拡張関係要素	produce,consume,assign,predecessor,succesor,setup,interval

(2) 開発モジュールの仕様

開発したモジュールは、Microsoft 社 Windows 95/98/NT/2000/NT/XP の各 OS 上で利用する。本モジュールはプログラムに組み込んで利用することを前提としており、利用にあたっては、VisualBasic、VisualBasic for Application、VBScript、VisualC++、などの開発言語と組み合わせて使う。IDL を生成することで、Java とのインタフェースも必要に応じて可能となる。

本モジュールは、表 2 に示す COM オブジェクトとして提供される。pslxi.dll は、アプリケーションプログラムから直接呼び出される関数が格納されており、主にデータの入出力や内部操作を行う。一方、pslxs.dll は、クライアントサーバ間の通信に用いるモジュールである。

表 2 提供するモジュール名と概要

pslxi.dll	PSLX 入出力インタフェースモジュール
pslxs.dll	PSLX クライアントサーバモジュール

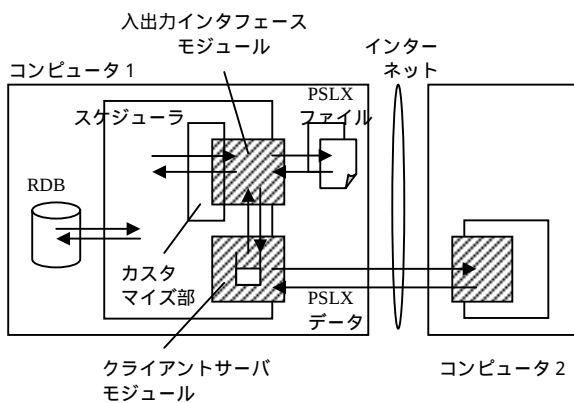


図 1 開発モジュールの概要

PSLX 入出力インタフェースモジュールは、PSLX の標準仕様に対応する XML ファイルを入力および出力するためのモジュールである。このモジュールに含まれる関数をアプリケーションプログラムが呼び出すことで、XML を意識せずに簡単な記述形式でデータのやりとりが可能となる。

PSLX 入出力インタフェースが、同一 PC 内にあるファイルを対象としているのに対し、PSLX クライアントサーバモジュールは、インターネットで接続された物理的に異なる PC 上にあるアプリケーション間で PSLX データを交換するためのものである。このモジュールに含まれる関数を利用して、異なるアプリケーションを簡単に連携させることができる。

(3) 商用スケジューラ用インタフェース

開発した 2 つのインタフェースモジュールを利用して、国内 4 社のスケジューリングパッケージソフトウェア向けのインタフェースを開発し、以下のように実装した[4]。

a) ASPROVA 用インタフェース

アスプロバ株式会社の提供するスケジューラである ASPROVA が、PSLX 形式のデータを読み書きでき、さらに遠隔地からのデータ入力や問合せに対して対応するためのインタフェースモジュールを開発した。ASPROVA 本体に対するプラグインモジュールとして定義し、内部で定期的にメッセージを処理する。

b) ACCROAD 用インタフェース

株式会社ロジックスジャパンのスケジューラである ACCROAD が、PSLX 形式のデータを読み書きでき、さらに遠隔地からのデータ入力や問合せに対して対応するためのインタフェースモジュールを開発した。ACCESS 上で独自のメニューをもち、RDB とスケジューラを制御する。

c) DIRECTOR 用インタフェース

株式会社シムトップスのスケジューラソフトウェア製品である DIRECTOR が、PSLX 形式のデータを読み書きでき、さらに遠隔地からのデータ入力や問合せに対して対応するためのインタフェースモジュールを開発した。サーバモジュールが待機し、進捗の受け付けや納期回答などを行う。

d) JoyScheduler 用インタフェース

ジェイティエンジニアリング株式会社のスケジューラソフトウェア製品である JoyScheduler が、PSLX 形式のデータを読み書きでき、さらに遠隔地からのデータ入力や問合せに対して対応するためのインタフェースモジュールを開発した。VisualBasic で作成し、RDB を介してスケジューラを遠隔操作する。

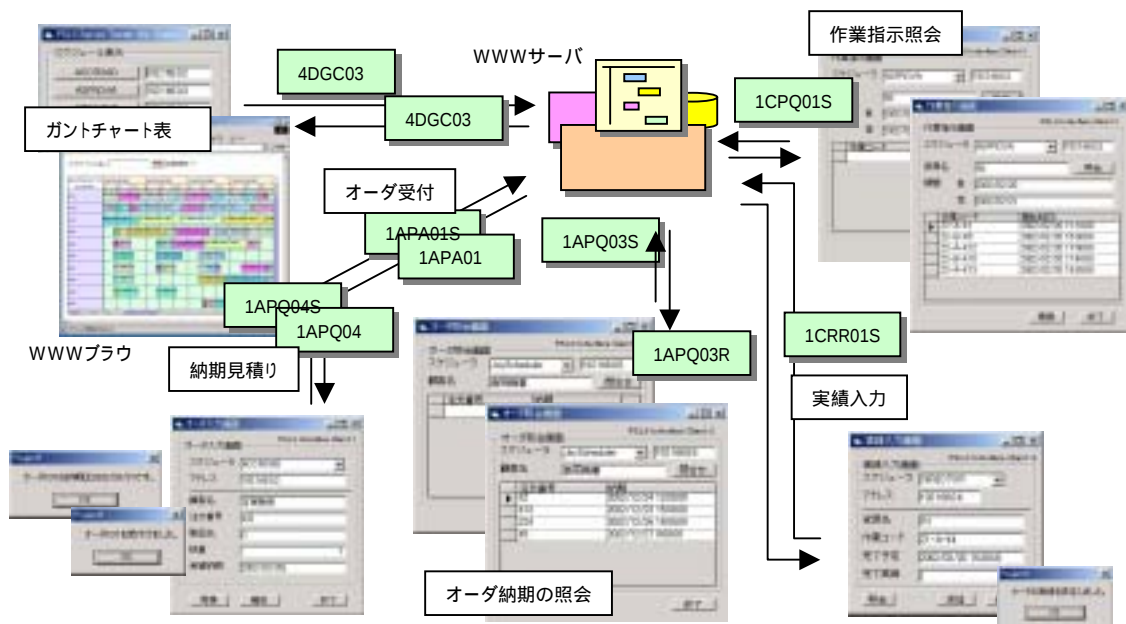


図2 スケジューラ用共通クライアント

(4) デモ用モジュール

スケジューラベンダー各社の製品に対して、共通のインタフェースでプログラムが連携できることをデモンストレーションするために、以下の6つのユースケースを設定し、それぞれに対応するクライアントプログラムを開発した。

a) ガントチャート遠隔表示

スケジューラが持っている現時点でのスケジュール内容を遠隔地でガントチャートの形式で表示させる。

b) オダ内容の照会

スケジューラが対象としている現時点でのオダの内容を特定の顧客について遠隔地から照会する。

c) 納期見積もり / オダ設定

スケジューラに対して新たなオダを設定する。あるいはオダを設定した場合の納期を照会する。

d) 作業指示照会

各作業場において当日分または翌日分の作業指示をスケジューラに問い合わせる

e) 生産実績入力

各作業場において当日分の生産実績をスケジューラに送信する

f) 再スケジューリング実施

遠隔地からスケジューラに対して再スケジューリングを行わせる

上記6つのプログラムは、(2)の2つのインタフェースモジュールを利用して VisualBasic を用いて開発した。(3)の商用スケジューラ用インタフェースとともに用いることで、図2に示すように実際の複数の商用スケジューラと連携が可能となった。

(5) WWW デモサイトの構築

PSLX 入出力インタフェースモジュールおよびクライアントサーバモジュールの機能を紹介するために、WWW 上でデモを行うプログラムを開発した。HTML ファイルおよび ASP ファイルからなる。構築したデモサイトは、サプライチェーンマネジメントを拡張した新しいビジネスモデルとしてもその活用が可能なように、以下のような工夫をほどこした。

取り上げたビジネスシナリオとして、4社の工場から

なるサプライチェーンによって、ひとつの完成品（電気自動車）が生産されるという設定とし、顧客はインターネットから注文を出すものとする。4社でチェーンが構成されているために、納期回答などの管理は単独のスケジューラでは行えない。

このような場面設定のもと、WWW サーバーが各スケジューラとの通信を行い、自動的にオダ設定や納期回答を開発したインタフェースモジュールを用いて行えるようにした。そして4社の工場はそれぞれ異なる商用のスケジューラが管理することで、スケジューラの連携が可能であることを示した。構築した WWW サイトでは顧客向け以外にも生産管理担当者や現場作業員向けのページがあり、このサイトを通して APS (Advanced Planning and Scheduling)を実現している[1]。

4. インタフェースの詳細

(1) インタフェース関数一覧

開発したインタフェースモジュールが持つ関数の種類は180以上になるため、ここでは個々の関数の説明は省略する。詳細の内容についてはインタフェース仕様書[5]を参照のこと。開発した関数は、以下のようなカテゴリに分類することができる。なお、末尾の数字はカテゴリに該当する関数の数である。

- a) 全体に関するメソッド : 3
- b) 外部起動関係 (サーバ用) : 8
- c) 外部起動関係 (クライアント用) : 6
- d) データの生成に関するメソッド : 8
- e) データの入出力に関するメソッド : 11
- f) データの入力補助 : 6
- g) 問題に関する設定事項 : 7
- h) 基本要素に対する情報の設定 (共通) : 20
- i) 表示関係の属性の設定 : 8
- j) 注文に関する情報の設定 : 15
- k) 作業や事象に対する情報 : 12
- l) 作業と他の要素との関係 : 9
- m) 在庫量に関する表現 : 5
- n) 負荷量に関する情報 : 4

- o) 制約情報の設定：1 6
- p) 制約情報の取出し：1 6
- q) 制約情報の種別：1 6
- r) クエリー用関数：1 1

(2) サンプルプログラム (ファイル入出力)

ファイル入出力のサンプルとして、Microsoft Excel のデータを VBA で読み込み、PSLX インタフェースを利用していったん XML ファイルに書き出した後に、PSLX 対応のガントチャートビューアで表示する例を紹介する。

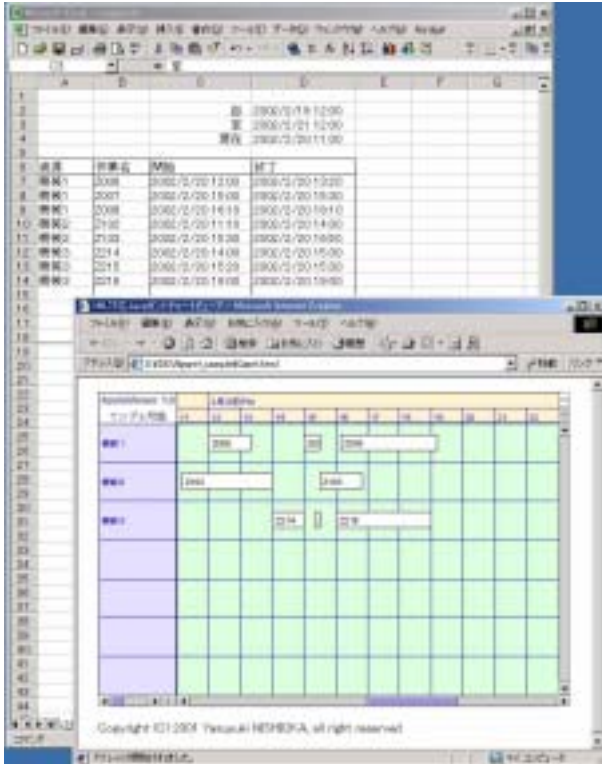


図3 XML ファイル出力の例

リスト1 PSLX 出力のためのプログラム

```
Sub OutSchedule()
    Dim pslx As PSLXInterface
    Dim ie As Object
    Set pslx = New PSLXInterface
    Dim p1 As Long, p2 As Long

    Dim i As Integer
    i = 7

    pslx.CreateProblem "サンプル問題"
    pslx.SetTimeHorizon Cells(2, 4), Cells(3, 4)
    pslx.SetTimeNow Cells(4, 4)
    pslx.SetViewScale 3600, Cells(4, 4)

    Do While Cells(i, 1) <> ""
        p1 = pslx.CreateOperation(Cells(i, 2))
        p2 = pslx.CreateResource(Cells(i, 1))
        pslx.SetDispName p1, Cells(i, 2)
        pslx.SetAssign p1, p2, 1
        pslx.SetOperationTime p1, Cells(i, 3), Cells(i, 4)
        i = i + 1
    Loop

    pslx.WriteData ThisWorkbook.Path & "\link.xml"
    Set ie = CreateObject("InternetExplorer.Application", "")

    ie.Width = 800
    ie.Height = 600
    ie.Navigate2 ThisWorkbook.Path & "\Gantt.html"
    ie.Visible = True

End Sub
```

図3は、Excel 上に設定された表と、その内容に対応するガントチャート表示結果である。Excel 上で作業の開始、終了時刻を設定し、VBA を起動することで、その都度、対応するガントチャートが表示される。このプログラムのソースコードをリスト1に示す。

(3) サンプルプログラム (クライアント/サーバ)

続いて、クライアントサーバ間の通信機能を用いたサンプルを示す。図4のサンプルは、クライアントとサーバ間の簡単なデータのやりとりを行うものである。クライアントは、特定の PSLX ファイルの内容をサーバに送り、サーバからのリプライを待つ。一方、サーバは、複数からのメッセージをスタックに蓄積していき、任意の順でリプライメッセージを送信元に発信する。

このプログラムの、サーバ側のプログラムのソースコードをリスト2に示す。このように、異なるアプリケーション間での PSLX データの送受信が数行のコーディングでできるようになった。

(4) メソッドの命名規則

PSLX に準拠した記述形態で XML を記述しアプリケーション間で情報交換する際に、単にデータを送るだけでなく Web サービスとして何らかの機能の実現を要求したい場合が多い。実際にスケジューラと連携する場合には、特にスケジューラに何をさせたいのかを明示的に示さなければならない。

リスト2 サーバ側プログラム

```
Dim ps As Object

' GetQueue ボタンが押された
Private Sub Command1_Click()

    Dim id As Long
    id = ps.GetMessageQueue
    If id = 0 Then Exit Sub

    GQRet.Text = id
    GQMsg.Text = ps.GetMessage(id)
    GQMsgType.Text = ps.GetMessageType(id)
    ps.LoadMessage id

End Sub

' Reply ボタンが押された
Private Sub Command3_Click()

    Dim id As Integer
    If GQRet.Text = "" Then Exit Sub

    id = CLng(GQRet.Text)
    ps.LoadData (GQReply.Text)
    ps.ReplyMessage GQMsgType.Text, id
    ps.ReleaseMessage id

End Sub

Private Sub Form_Load()

    Set ps = New PSLXInterface
    Dim ret As Integer

    ps.InitProblem
    ret = ps.StartMessageServer(950)

End Sub

Private Sub Form_Unload(Cancel As Integer)

    ps.StopMessageServer
    Set ps = Nothing

End Sub
```

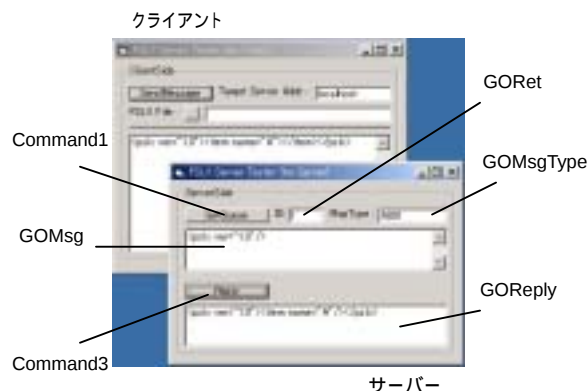


図4 クライアントサーバ通信の例

そこで、メッセージ名として以下のような記述ルールを設け、PSLX コンソーシアムの仕様にも組み込む形で整備した。図2に示す7桁の文字列は、このメッセージ名を表している。具体的には、2桁の種別番号、2桁の区分番号、2桁のユースケース番号、そして1桁のメッセージタイプからなる。

まず、種別番号の内容は、表3のようにトランザクションに関するメッセージ、マスタ情報に関するメッセージ、統計値や集約値に関するメッセージ、そしてその他の制御や管理で利用するメッセージなどがある。また、区分番号はP(計画)、R(実績)、C(その他)から一文字、Q(問合せ)、A(追加)、R(修正)、D(削除)の中から1文字の計2文字から構成される。さらに、ユースケース番号は、00から99までの番号で、製造業のさまざまなユースケースを表現し、最後のメッセージタイプは、S(送信)、R(返信)、E(例外)などの文字となる。

表3 メッセージ分類とコード

区分	コード
トランザクション情報	受注:1A, 発注:1B, 作業:1C, 工程:1D, 品目:1E, ロット:1F, 資源:1G, カレンダー:1H, 全体:1J
統計集約情報	利益:2A, 収入:2B, 支出:2C, 操業度:2D, 生産数:2E, オーダ数:2F, 在庫数:2G, 負荷数:2H
マスター情報	顧客:3A, サプライヤ:3B, 工程:3C, 品目:3D, 設備:3E, BOM:3F, 手順:3G, 制約:3H, 指標:3J, コスト:3K
制御/管理情報	受注:4A, 発注:4B, 計画:4C, スケジューラ:4D, MES:4E, 労務:4F, 在庫:4G, PDM:4H, 物流:4J, 会計:4K

5. ビジネス評価結果

今回開発した生産スケジューラ用インタフェースモジュールの評価として、技術面のみでなくビジネス的な側面からの考察が重要となる。なぜなら、より多くのベンダーやユーザーがこのモジュールを利用し、PSLXの仕様がデファクトスタンダードとして確立されなければ、技術的にどれだけ優れていたとしても結果的には使われないからである。

以下に、今回のプロジェクトに参加したスケジューリングベンダー4社に対して、プロジェクト終了後にアンケート調査を行った結果の意見を抜粋する。

- (1) PSLXの表現力は非常に高いのでパッケージをこれに併せることは可能。
- (2) スケジューラベンダーが協力して接続をおこなった点は高く評価できる。
- (3) 大量データのパフォーマンス、エラー処理、信頼性に対するさらなる検討が必要。
- (4) 技術情報の標準化はかなりできたが交換手順がまだ未完成。
- (5) 標準化による開発工数削減により販売数量が増える可能性はある。
- (6) スケジューラがより身近になることでスケジューラ市場全体の底上げ効果が期待できる。
- (7) 普及に際してはMESやERPとの連携が非常に重要。
- (8) トラブル時のサポート体制、将来的な機能拡張の体制などの整備が必要。
- (9) スケジューラベンダーとしては今後もPSLXには深くかかわる予定。
- (10) ユーザからのプレッシャーを利用して採用を周辺のベンダーに広げたい。
- (11) 実用化するには、ある程度ターゲットをしばって詳細をつめる必要あり。
- (12) 実用化のためには早急に実証プロジェクトを立ち上げる必要あり。

このように、本プロジェクトの成果物であるPSLXインタフェースモジュールは、今後の製造業のIT化を支えるひとつの要素技術として発展する可能性を多分に秘めており、これからの継続的な開発の努力により大きな効果を生み出すことが期待できる[6]。

6. 参加企業及び機関

本プロジェクトは、開発内容の一部を再委託するという形式で以下の企業の協力を受けた。

- (1) アスプローバ株式会社
- (2) ジェイティエンジニアリング株式会社
- (3) 株式会社シムトップス
- (4) 株式会社ロジックスジャパン
- (5) 有限会社アブストウェブ

7. 参考文献

- [1] 西岡靖之: APS 先進的スケジューリングで生産の全体最適を目指せ、日本プラントメンテナンス協会、東京(2001)
- [2] PSLX コンソーシアム、<http://www.pslx.org/>
- [3] PSLX コンソーシアム: PSLX 技術仕様書(勧告候補版)、<http://www.pslx.org/> (2002)
- [4] 生産スケジューリングライブラリー: APS 市販パッケージ調査、<http://www.img.k.hosei.ac.jp/pslib/survey2002/index.htm>
- [5] 西岡靖之、PSLX インタフェース仕様書、<http://www.pslx.org/> (2002)
- [6] 西岡靖之、部門間・企業間での情報共有一元化とリアルタイムな情報共有に向けて、日経デジタルエンジニアリング、Vol.53、pp.122-125 (2002)