

SASと外部ソフトウェアの連携

Windows GUI構築とExcelデータの利用

○高田一行¹、迫田英之¹、阿部一也¹、村松徳之¹、宮原秀夫¹、木村範昭²
(¹ 株式会社タクミインフォメーションテクノロジー、² 専修大学 商学部)

要旨：

Windows GUIと外部ソフトウェアをSASと連携させ、環境に依存しない使いやすいWindows GUIメニューを利用することにより、複雑なレイアウトのEXCELシートデータをSASデータセットに変換し、SASの分析機能を利用するまでの方法を紹介する。

キーワード：SAS Integration Technologies, PowerShell, WPF(Windows Presentation Foundation), Excel連携

SAS Integration Technologiesとは

SAS

SAS Integration Technologies



シームレスな連携を提供

外部環境

「別なor同じ」マシンの
SAS,Excel,VB,DB等

SAS Integration Technologiesでの外部連携

外部連携 2つのメリット

SASの情
報を
有効利用

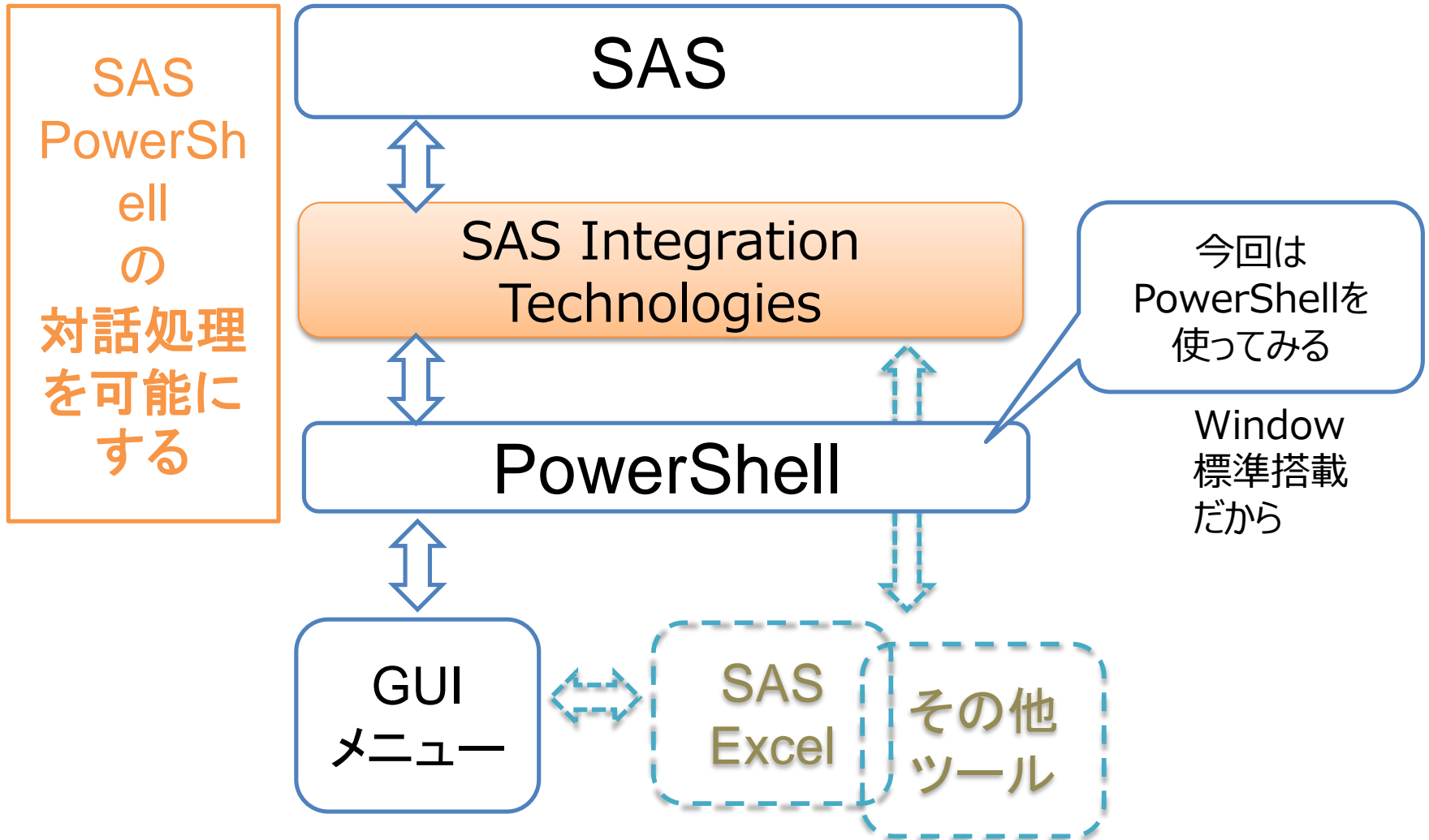
SASで情
報を
有効利用

SAS
ユーザー総会
2017

SAS Integration Technologiesを用いて

SASの情報を有効利用

SASと外界のシームレス連携



ローカルPC環境での Integration Technologies

SASWorkspaceクラスを使用
(SAS Integration TechnologiesのIOM機能使用)

主要なプログラムから利用が可能

汎用化(関数化など)すると以外に簡単

PowerShellとは

Windows用のシェル言語

Cmd.exe
いわゆるDOSコマンド機能

Windows7以降標準

スクリプト言語

.NET機能フルサポート

処理関数群(Cmdlet)

大容量データには向かない

ユーザー関数作成機能

速度は速くない

コーディング例をしてみる

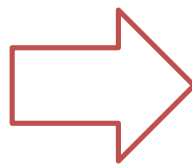
PowerShellからExcel操作
Excel起動⇒ファイル指定⇒シート名取得

Cmdlet

```
$excel=New-Object -ComObject Excel.Application  
$book=$excel.Workbooks.Open("C:¥Temp¥データ.xlsx")  
$names=$book.Sheets | Select-Object -Property Name
```

変数名

\$nameの内容



クラス
授業アンケート
月別納豆消費

PowerShellでSASを扱う

たとえばこんな処理

SASプログラムをインタラクティブに実行

結果（データ、レポート、ログ）受け取り

様々な表示へ（メニュー化、Excelへ）

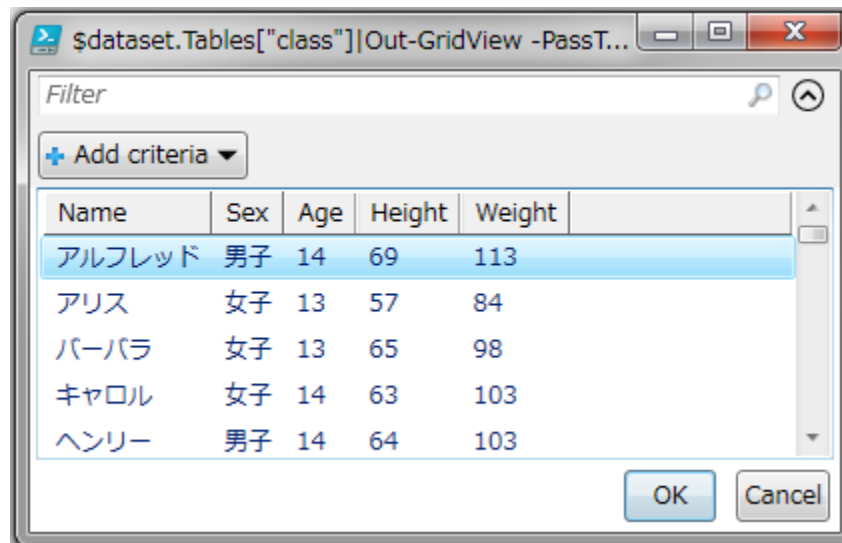


プログラムを汎用（簡易）化できる

汎用化した使用例

SASHELP.CLASSをPowerShellで取得・表示

```
start-sas  
$sasSource="select * from sashelp.class"  
get-sasdata -query $sasSource -table class  
$dataset.Tables["class"] | Out-GridView -PassThru
```



| Name | Sex | Age | Height | Weight |
|--------|-----|-----|--------|--------|
| アルフレッド | 男子 | 14 | 69 | 113 |
| アリス | 女子 | 13 | 57 | 84 |
| バーバラ | 女子 | 13 | 65 | 98 |
| キャロル | 女子 | 14 | 63 | 103 |
| ヘンリー | 男子 | 14 | 64 | 103 |

Window(GUI)メニューの作成

PowerShellを介してWPFでメニュー作成

使用するGUI

WPF(Window Presentation Foundation)

画面デザインとロジックの分離 (コードビハインド)
新たなGUI (UWP) への橋渡し
データバインド (データ受け渡機能の一種)
綺麗なデザイン
* 従来のFormアプリケーションも可

WPFでのWindow表示プログラム

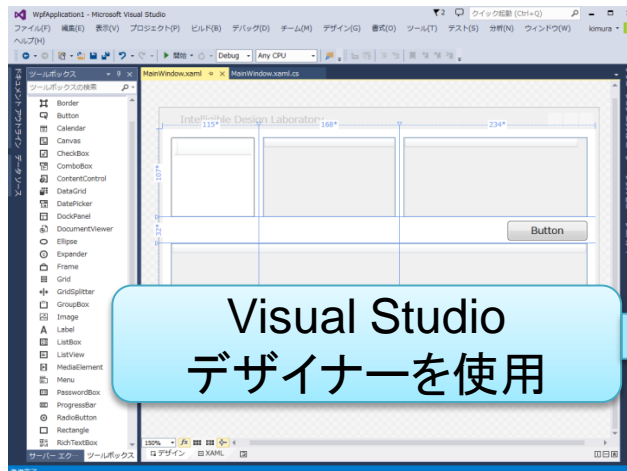
画面定義例（空のWindow表示）

```
Add-Type -assemblyName PresentationFramework
[xml]$xaml = @"
<Window
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <!-- ここにコンポーネント定義を追加 -->
</Window>
"@
$reader=(New-Object System.Xml.XmlNodeReader $xaml)
$Window=[Windows.Markup.XamlReader]::Load( $reader )
$Window.ShowDialog()
```

XAML（ザムル）という書式で画面定義（赤文字部分）
Visual Studioのデザイナーも使用できる

WPFでWindow作成

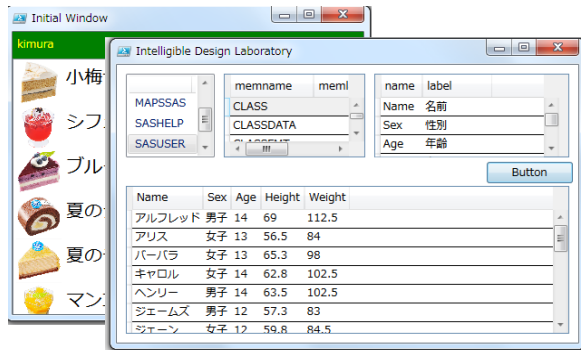
画面作成は簡素化できる（作成法2種類）



```
<Grid>  
<Grid.RowDefinitions>  
  <RowDefinition Height="107*" />  
  <RowDefinition Height="32*" />  
  <RowDefinition Height="181*" />  
</Grid.RowDefinitions>  
<Grid.ColumnDefinitions>  
  <ColumnDefinition Width="115*" />  
  <ColumnDefinition Width="160*" />  
  <ColumnDefinition Width="160*" />  
</Grid.ColumnDefinitions>
```

テキストエディター
を使用

```
<ListView.view>  
<GridView>
```



SASデータと合わせ
PowerShellで
データ処理 & 表示

SAS
ユーザー総会
2017

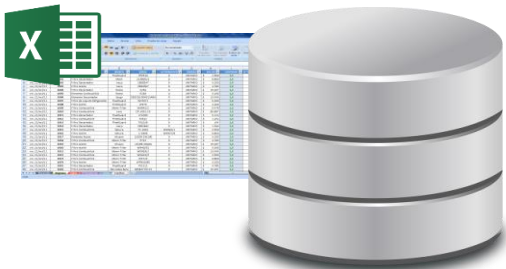
SASで情報を有効利用

使われないExcel過去資産

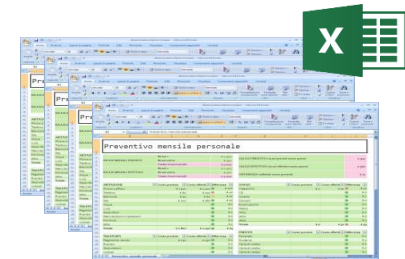
謳い文句: DB化しBIで有効利用

現実

プレーンなExcelデータ
今後新たに発生するデータ



複雑なExcelレイアウト
倉庫入り!



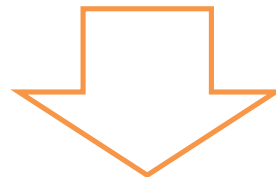
一般的なExcelデータの取り込み

単純な表形式なら

SAS Importプロセス
CSVにして読み込み

少量のブックなら

DDEで直接読み込み



複雑なレイアウトだと？

大量のブックは？

SASだと出来るExcel資産**有効**利用1

Excelインポートは結構めんどろ

シートを見ると

自由度高い
レイアウト

セル内改行

セル結合

ブックを見ると

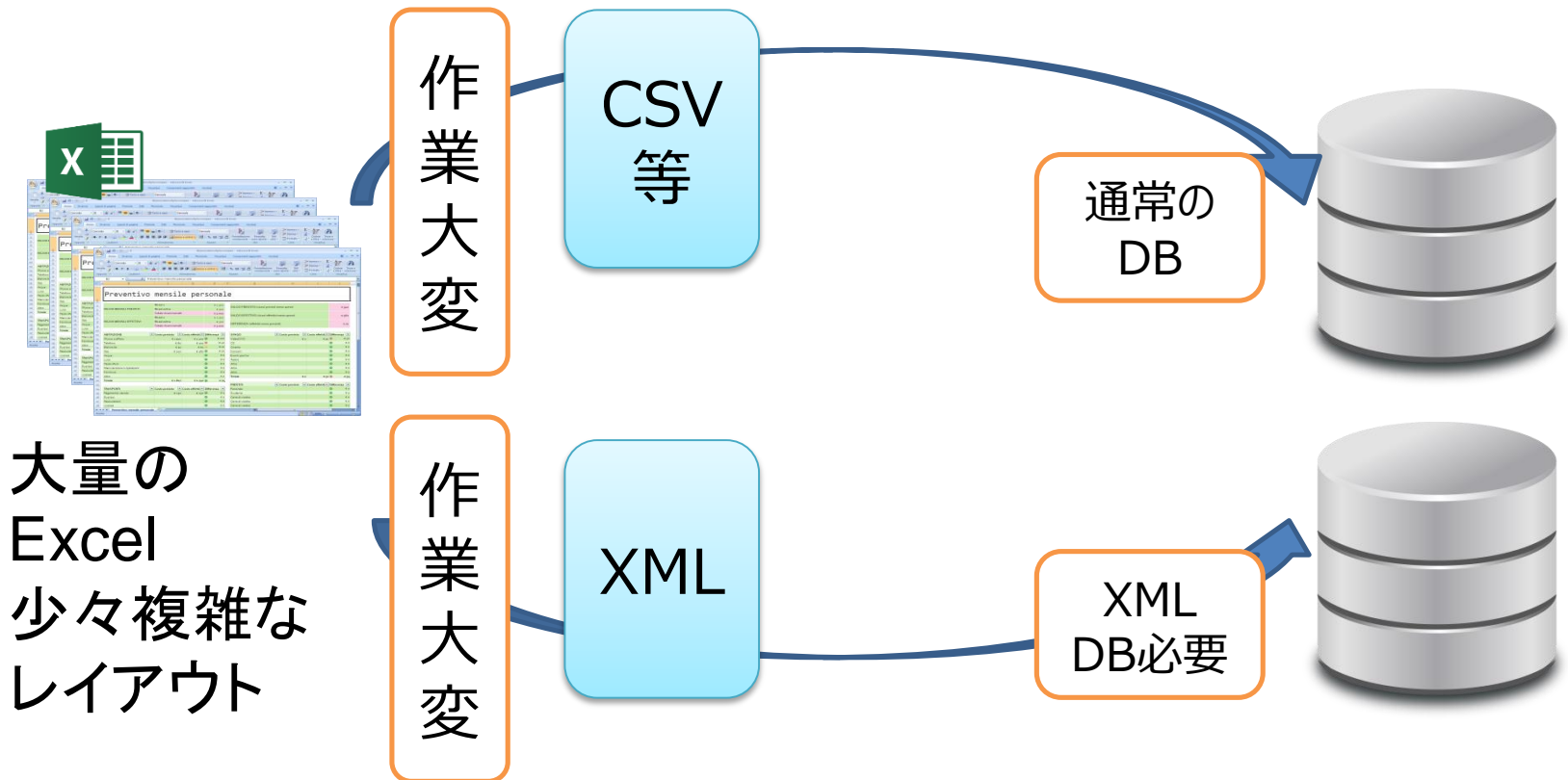
大量のブック

Open、Close

バージョン非互換

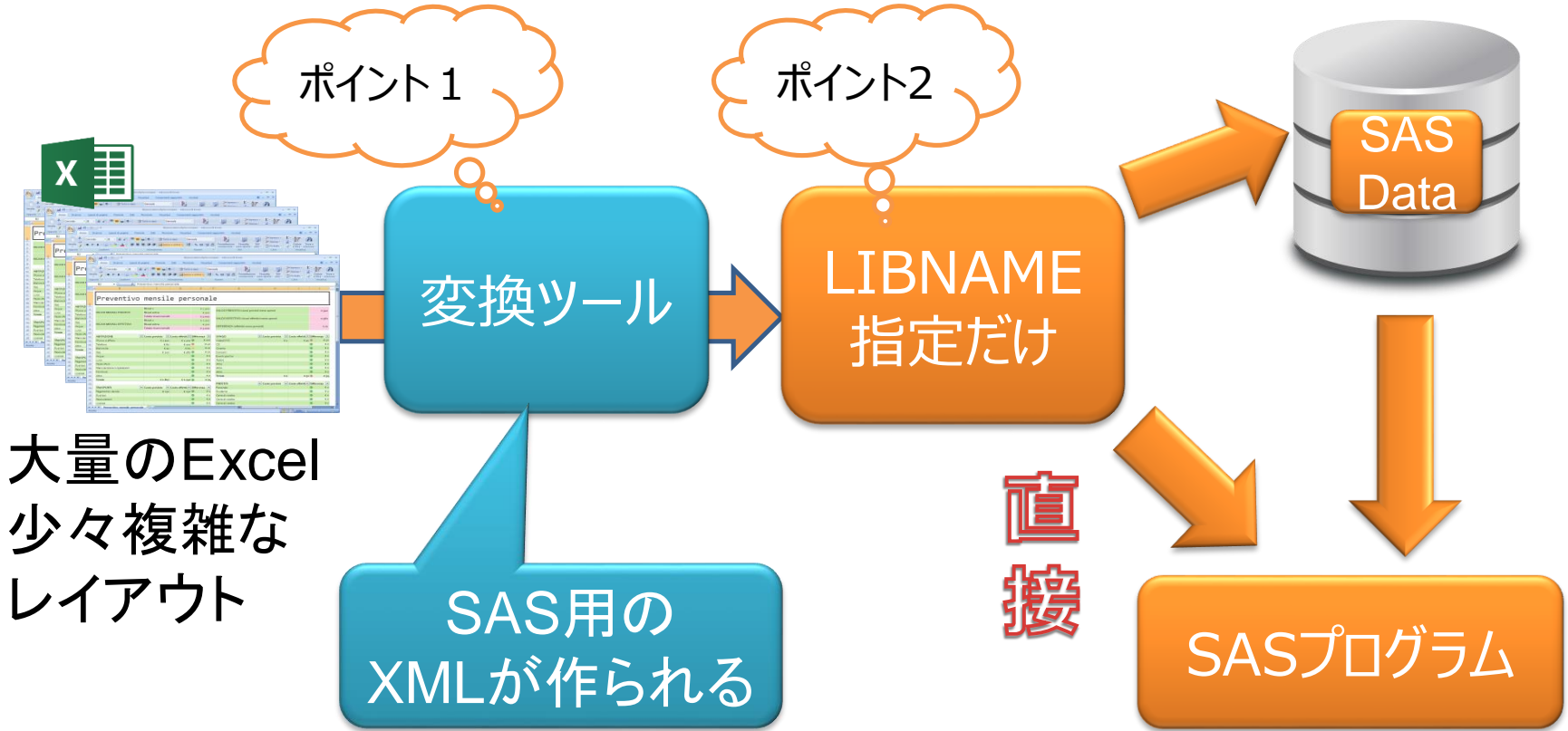
SASだと出来るExcel資産**有効**利用2

今まで嫌った理由



SASだと出来るExcel資産有効利用3

新たな仕組みの提言



変換ツールのメリット1

変換と同時にSASライブラリエンジンで利用

エネルギー使用量の精算計算表
工場-事業場-工

| エネルギーの種類 | 使用量 | | 換算係数 | | | | | |
|-------------|-------------------|-----------|-------------|--------|-------|-------|---------|------|
| | 単位 | 数値 | 数値 | 単位 | | | | |
| 燃料 | 原油 | kl | 20 | 784 | 38.2 | GJ/kl | | |
| | 原油のうちコンデンサート(NGL) | kl | 21 | 741 | 35.3 | GJ/kl | | |
| | 揮発油(ガソリン) | kl | 22 | 761 | 34.6 | GJ/kl | | |
| | ナフサ | kl | 23 | 773 | 33.8 | GJ/kl | | |
| | 灯油 | kl | 24 | 881 | 36.7 | GJ/kl | | |
| | 軽油 | kl | 25 | 943 | 37.7 | GJ/kl | | |
| | A重油 | kl | 26 | 1,017 | 39.1 | GJ/kl | | |
| | B・C重油 | kl | 27 | 1,131 | 41.9 | GJ/kl | | |
| | 石油アスファルト | t | 28 | 1,145 | 40.9 | GJ/t | | |
| | 石油コークス | t | 29 | 887 | 29.9 | GJ/t | | |
| | ガス | 石炭ガス | 液体石炭ガス(LPG) | t | 30 | 1,524 | 50.8 | GJ/t |
| | | 石炭ガス | 石炭液化ガス(LPG) | t | 31 | 1,392 | 44.9 | GJ/t |
| | | 可燃性 | 液体石炭ガス(LPG) | t | 32 | 1,747 | 54.5 | GJ/t |
| | | 天然ガス | その他可燃性ガス | t | 33 | 1,438 | 43.5 | GJ/t |
| | | 石炭 | 原料炭 | t | 34 | 988 | 29.0 | GJ/t |
| | | | 一般炭 | t | 35 | 900 | 28.7 | GJ/t |
| | | | 無煙炭 | t | 36 | 968 | 28.9 | GJ/t |
| | | | 石油コークス | t | 37 | 1,088 | 29.4 | GJ/t |
| | | | コールタール | t | 38 | 1,417 | 37.3 | GJ/t |
| | | | コークス炉ガス | t | 39 | 823 | 23.2 | GJ/t |
| | 高炉ガス | | t | 40 | 138 | 3.4 | GJ/t | |
| | その他の燃料 | 低炉ガス | t | 41 | 345 | 8.4 | GJ/t | |
| | | 都市ガス 1.2A | t | 20 | 1,000 | 50.0 | GJ/t | |
| | | その他 | t | 0 | 0 | 0 | GJ/t | |
| | 電気 | 産業用電気 | CJ | 10 | 10 | 1.02 | (換算係数) | |
| 産業用以外の電気 | | CJ | 20 | 27 | 1.36 | | | |
| 温水 | | CJ | 20 | 27 | 1.36 | | | |
| 冷水 | | CJ | 0 | 0 | 0 | | | |
| 小計① | | | | 22,849 | | | | |
| 小計② | | 一般電気事業 | 千kWh | 30 | 290 | 9.97 | GJ/千kWh | |
| | | 夜間電気 | 千kWh | 10 | 93 | 9.28 | GJ/千kWh | |
| | | 上層以外の電気 | 千kWh | 50 | 488 | 9.76 | GJ/千kWh | |
| | | 自家発電 | 千kWh | 10 | 93 | 9.28 | GJ/千kWh | |
| 小計③ | | 千kWh | 90 | 880 | | | | |
| 合計④ (①+②+③) | | | 23,729 | | | | | |
| 原油換算 | kl | | 812 | 0.028 | GJ/kl | | | |

第一種エネルギー-量産指定工場
第二種エネルギー-量産指定工場

単位なし 5,000 M以上
単位なし 1,000-5,000 M
単位なし 500 M未満

注) 約定書は、工場-事業場ごとの約定の名称、種別(第一種、第二種の区分)を明示します。

変換
ツール

```
c:¥temp
sample.xml
energy.xml
dfile1.xml
.....
```

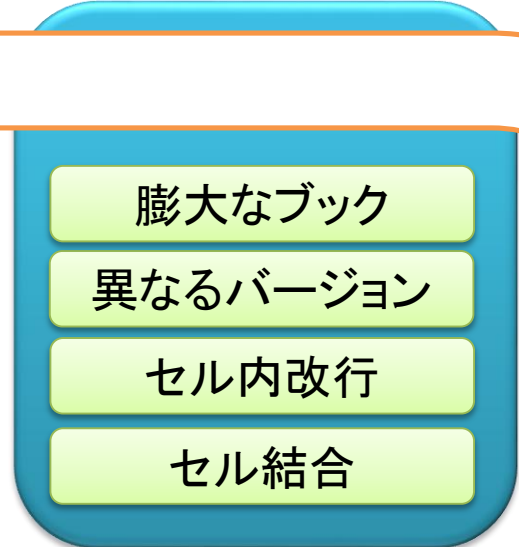
```
libname xlib xlv2 "c:¥temp";
proc print data=xlib.energy;

SASシステム
```

変換ツールのメリット2

大量かつ異なるバージョンbookもSAS用のデータに

prod1995_05.xls
prod1995_06.xls
.....
prod2000_01.xls
prod2000_02.xls
.....
prod2017_04.xlsx
prod2017_05.xlsx
prod2017_06.xlsx



c:¥temp
prod1995_05.xml
prod1995_06.xml
.....
prod2000_01.xml
prod2000_02.xml
.....
prod2017_04.xml
prod2017_05.xml
prod2017_06.xml

```
libname xlib xmlv2 "c:¥temp¥prod:";  
proc print data=xlib.energy;
```

SASシステム

眠っていたExcel資産の有効利用 **だけでなく**

シート上のデータをそのままSASデータとして利用できるため、過去データだけでなく現在利用されているExcelを、他システムに移行せず、そのまま生かした真にシームレスな環境の提供が可能と思われる。

```
graph TD; A[ ] --> B[SASを使うと]; B --> C[日夜で生まれる Excel資産をそのまま SASデータとして 利用・提供可能];
```

SASを使うと

日夜で生まれる
Excel資産をそのまま
SASデータとして
利用・提供可能

まとめ

今回の調査ではSASと外部機能をシームレスに連携されることにより、SASのさらなる可能性を発見することができた。画面構築も思ったよりも難しいものではなく、ユーザー部門のパワーユーザーであれば作成することが可能であろう。

特にExcelで構造が複雑化したデータの活用は多くの企業やSASユーザーにとってもジレンマとなっており、これを解決することはSASの優位性をさらに高めることにもつながる。

現在、データの再利用の面からExcelの利用を控え、Web等の入力に切り替える企業も増えているが、エンドユーザーの利用面から見ると、不便に感じる部分もあり、Excelが活用できるのであればその方が優位であると考える。

過去のデータ資産の活用、そしてこれからのデータ資産の運用の観点を含めて述べても、汎用性と解析処理能力はビックデータを扱う多くの企業においても求められるところであり、この点においても今回の研究を活用すればSASユーザーにとって新たなアプローチからソリューションを行うことができる。

お問い合わせはこちらへ！

株式会社タクミインフォメーションテクノロジー

東京都豊島区池袋2-40-13池袋デュープレックスB's 11F
<http://www.takumi-it.co.jp>

TEL: (03)5979-7381

Email: contact@takumi-it.co.jp