

カスタマイズ可能な Web サービス機能を利用した SynViz 協業ビジネスの実現

Developing a Collaborative Business Model with SynViz as a customizable Web Service

(株)日立東日本ソリューションズの製品であるプロジェクト管理ツール SynViz は、現行製品がカスタマイズ性強化のニーズの高まりに対応できていない問題があった。この課題を根本的に解決するため、われわれは Web API とカスタマイズ機能を備えた次期 SynViz の製品基盤を研究試作した。

Web API は REST アーキテクチャを採用したことで、ネットワーク経由で SynViz のデータにアクセスすることができる。クライアントを選ばずデータ連携バッチ処理などさまざまな用途に使用できる汎用性を備えるとともに、後方互換性を維持し製品本体がバージョンアップしても Web API を利用したプログラムは変更なしに利用できる設計とした。

カスタマイズ機能はモディフィケーションの必要なく、ユーザ定義のデータ項目や独自に開発した画面を追加する仕組みを提供する。

これらの成果により、一度作成したカスタマイズ機能をノウハウという形で蓄積し、お客様に安く早くよいソリューションを提供できる素地ができた。今後はビジネスパートナーとの連携を強化し、新しい SynViz を核とする協業ビジネスを推進していく。

庄子 真海 Shoji Masami
門司 太郎 Monji Taro
竹本 宗弘 Takemoto Munehiro
内海 宏律 Utsumi Hironori

1. 背景と動機

これまで、企業でのプロジェクト管理はマネージャによる経験や感覚に基づく属人的な手腕に頼る部分が大きかった。これを体系化・標準化し組織的なプロジェクト管理を実現する一助として、プロジェクト管理ツールの導入が進んでいる。

(株)日立東日本ソリューションズ(日立 TO)の SynViz シリーズはガントチャートを活用した工程・進捗管理を重点においたプロジェクト管理の見える化と標準化を実現する Web アプリケーションである。

業種を特定しない汎用的なプロジェクト管理ツールは市場に多種存在するが、SynViz シリーズは、ターゲットとするユーザの業種・業務を明確にし、個別ニーズに適合する機能を提供することをコンセプトとして掲げて

いる。基本機能を持つパッケージ製品基盤を核として、それに業種・業務に固有な追加機能を組み合わせて全体として効果の高いソリューションを提供する考え方である。

われわれは、このコンセプトを実現するまでのビジネス展開計画として、次の 4 つの段階を設定している。

第 1 段階: パッケージ製品を核としたカスタマイズによるソリューション開発を行う。その過程で業種・業務固有のノウハウを獲得する。

第 2 段階: 得られたノウハウを蓄積しオプション機能として開発することで、カスタマイズ開発の大幅な低価格化と運用開始までのスピードアップを実現する。

第 3 段階: ビジネスパートナーと連携し、ノウハウを洗

練させるとともに、ビジネスパートナーが持つ製品との付加価値のより高い「連携ソリューション」によりビジネスを展開する。

第 4 段階: 蓄積したノウハウから業務によらず効果の高い部分を製品本体にフィードバックして、販社などビジネスパートナーを通じた製品パッケージ単体のライセンス販売を拡大する。

ここで言うノウハウとは、その業種でのプロジェクト管理で利用されるプロセスや管理指標の他、ツールの機能、画面レイアウト、操作性など付加価値となるものを示す。

このようなビジネス展開計画のもと、ターゲット業種の展開に合わせて SynViz/PJ をアプリケーション基盤として次のようなシリーズ製品をリリースしてきた。

- 保守作業要員管理向け: SynViz/MP (2009 年発売)
- 情報システム開発向け: SynViz/SD (2010 年発売)
- 自動車部品メーカー設計業務向け: SynViz/EP (2011 年発売)
- 個別受注生産業向け: 個別受注生産テンプレート (2011 年リリース)
- 住宅メーカー向け: 案件管理テンプレート (検討中)

これらのシリーズ製品は展開計画の第 1 段階での核にあたるものであり、リリース後に個別にノウハウの蓄積・洗練を進めていくことを目指していた。

しかし、第 2 段階のノウハウの蓄積やオプション機能としての活用がなかなか進まない状況が続いた。その原

因を分析したところ、個別ニーズに対応する機能のカスタマイズ開発を行う際、想定していたよりも多くの場合でソースコードレベルの改修 (モディフィケーション) が発生していたためであった。モディフィケーションで作った機能は対象とした製品バージョンのソースコードに強く結合し、その後の再利用が困難である。

われわれは、当初目指した形でビジネス展開計画を進められるようにするため、製品本体の設計を根本的に見直し次期 SynViz 製品基盤(以下、コードネームである S2 と呼ぶ)の開発を目的として 2010 年 11 月から研究試作を開始した。

本稿ではその試作成果と製品化後のビジネス展望について述べる。

2. Web API

一般に、あるアプリケーションの機能を外部から呼び出すための関数群や規約を API (Application Program Interface) と呼ぶ。

カスタマイズ性を強化するため、S2 の API (以下、Web API) は REST アーキテクチャスタイルを採用した。

2.1 REST による API 設計

REST (Representational State Transfer) とは、Web システムに適したいくつかの設計原則を指す¹⁾。Amazon web services や Yahoo! などの著名な Web サービスで REST による API が採用されている^{2,3)}。

REST では API で扱われる対象をリソースと呼び、各リソースに対して一意のアドレス (URI) を設定する。

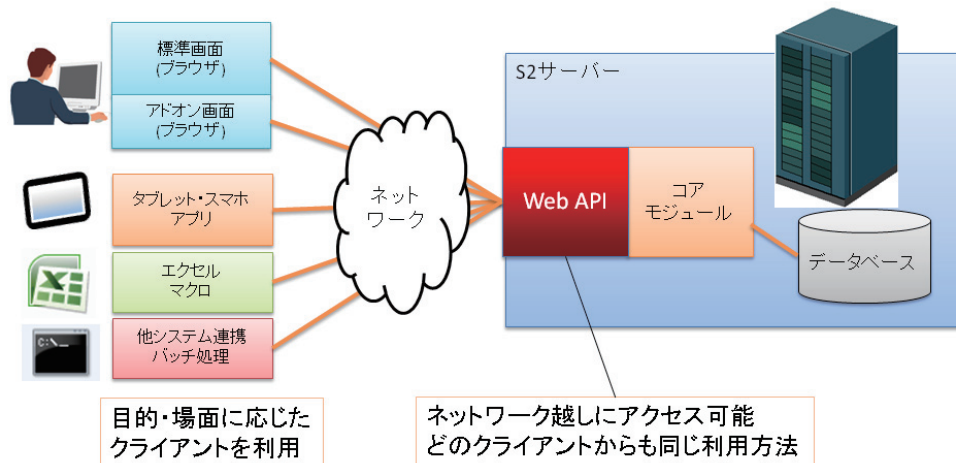


図 1 Web API の利用イメージ

さらに各リソースに対して CRUD 操作（作成，取得，更新，削除）を定義する。そして，CRUD の各操作を HTTP の POST, GET, PUT, DELETE の各メソッドにマッピングする。

Web API は，標準の画面であるブラウザを始めとするあらゆるクライアントに対し，タスク，アクティビティといった SynViz が取り扱う計画情報に関するオブジェクトモデルへの CRUD 操作を行う共通的なアクセス手段を提供する。（図 1）

たとえばプロジェクト ID（プロジェクトを特定するキー）が p001 であるプロジェクトに対して以下のように URI を指定して GET メソッドを実行すると，プロジェクトの全属性が JSON⁴⁾形式のテキストデータとして取得できる。

```
GET http://s2app/api/projects/p001
```

同様に，p001 プロジェクトを削除する場合は同じ URI に対し DELETE メソッドを実行する。

```
DELETE http://s2app/api/projects/p001
```

次に，p002 プロジェクトのすべてのタスク（作業項目）を取得するには，tasks という集合を GET する。末尾の rev=0 は，取得対象となる版番号を指定している。0 は最新版を意味する。

```
GET http://s2app/api/projects/p002/tasks
?rev=0
```

REST は，SOAP⁵⁾のような他の方式に比べて単純かつ軽量であり，クライアントの開発が容易であるという利点がある。HTTP による通信が可能であればどんな言語やフレームワークからでも利用できる。たとえば図 2 に示すように，エクセルのマクロを使用して直接 S2 サーバにアクセスし，ワークシートに作業予定をダウンロードした後，進捗入力した結果をワンクリックでサーバに反映するということが可能である。この例では CSV データのような中間ファイルを必要としないため，エンドユーザは普段と変わらないエクセルの操作感覚で S2 を利用できる。

またメソッドが基本的にリソースへの CRUD で構成



図 2 エクセル連携の例

されるため，一貫性と検索性に優れている。タスクやアクティビティなどのプロジェクトデータを構成するリソースの構造を理解すれば，カスタマイズエンジニアは容易に API を利用できる。

2.2 後方互換性

後方互換性とは，新しい機能を持つ製品が過去の機能を包含するように設計され，過去の製品と同じように動作可能であることを指す。

S2 は Web API は後方互換性を維持する設計とした。たとえば将来のバージョンで，あるメソッドの機能が一部追加された場合でも，これまでと同じ呼び出し方の場合には，以前の動作と変わらないようにする。追加オプション指定がされた場合にだけ新しい機能が実行されるようにする。S2 では REST スタイルを採用したことでこのような機能追加の影響を受けにくい設計を実現することができる。

後方互換性を維持することによって，Web API を利用して開発されたカスタマイズ機能やアプリケーションは，S2 本体がバージョンアップしてもそのままの形で利用し続けることができる。

3. S2 のカスタマイズ機能

S2 はモディフィケーションせずにカスタマイズを実現する手段を提供する。

3.1 カスタム属性機能

カスタム属性機能とは，タスクやアクティビティなどの S2 のオブジェクトに対してユーザ定義の属性を追加する機能である。属性の追加定義はカスタム属性編集画

面から行うため、運用管理者やカスタマイズエンジニアの手を借りることなく、エンドユーザが自ら操作して属性を追加できる。

カスタム属性は次のような型をサポートしており、項目ごとに目的に応じて使用する型を指定できる。

- 文字列(単一行, 複数行)
- 整数, 小数
- 日付, 日付時刻
- 列挙値 (任意の数の選択肢から一つ選択)
- 色
- プロジェクトメンバー/リソース

図 3 にカスタム属性の使用例を示す。この例では、タスクに対し「レビュー担当者」という表示名のカスタム属性を追加している。すると、ガントチャート画面のタスク情報として「レビュー担当者」の値を参照・編集できるようになる。

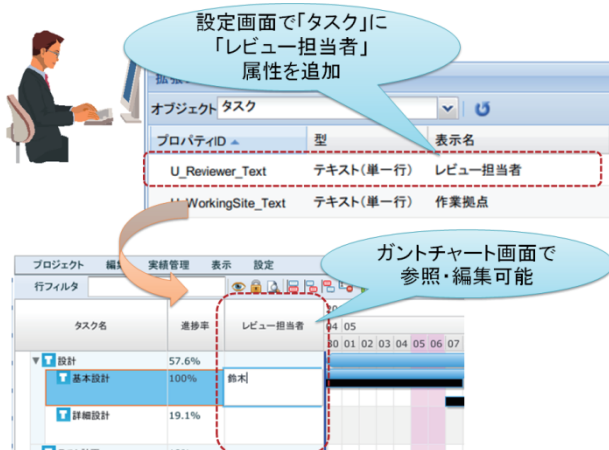


図 3 カスタム属性の使用例

追加されたカスタム属性は、Web API を通じて既定の属性とまったく同じように取得できる。たとえばタスクに対して Web API で GET メソッドを呼び出すと、タスクの既定の属性すべてに加え、レビュー担当者の属性も自動的に追加された JSON オブジェクトが取得される。

3.2 アドオン機能

アドオンとは、システム提供またはカスタマイズエンジニアが独自に開発したオプション機能のことで、1 個の独立したタブ画面として提供される。ユーザは、設定画面から必要なアドオンを選択することで、システム全体または自分が管理しているプロジェクトにアドオンを

追加できる。逆に不要なアドオンは使用しないように設定すれば、タブ自体が表示されなくなる。その様子を図 4 に示す。

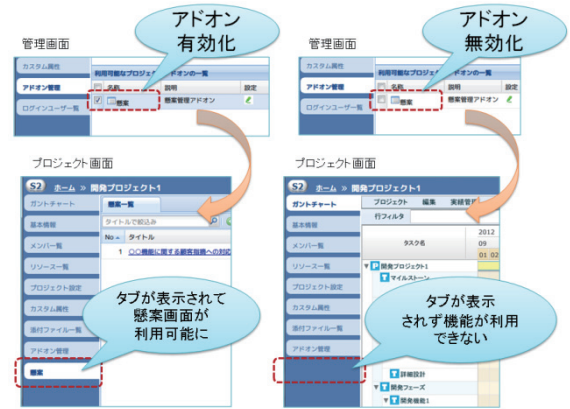


図 4 アドオンの有効・無効切り替え

S2 の標準提供機能では要求業務に対応できない場合、カスタマイズエンジニアがアドオンを開発することで対応する。

アドオン開発が必要になった場合、カスタマイズエンジニアはまずアドオンのひな形ファイル一式(開発ライセンス版に付属)を準備し、必要なファイルやソースコードを追加してアドオンを開発する。データの取得と更新はクライアント側(すなわちブラウザの JavaScript プログラム)から Web API を使用する。S2 機能へのアクセスはすべて API を通じて行うため、製品本体がバージョンアップしたときでも、API の後方互換性が維持されていれば、アドオンはそのまま移行可能である。

3.3 ダッシュボードとウィジェット

ダッシュボードとは、ポータルサイトによく見られるような、ウィジェットと呼ばれるミニアプリケーションを画面上に複数配置して表示した画面のことである。通常、ウィジェットはエンドユーザが追加・削除したり、ドラッグ&ドロップで配置を変えたりすることができる。S2 のダッシュボード画面の例を図 5 に示す。赤い枠が一つのウィジェットで、複数のウィジェットが配置されていることがわかる。

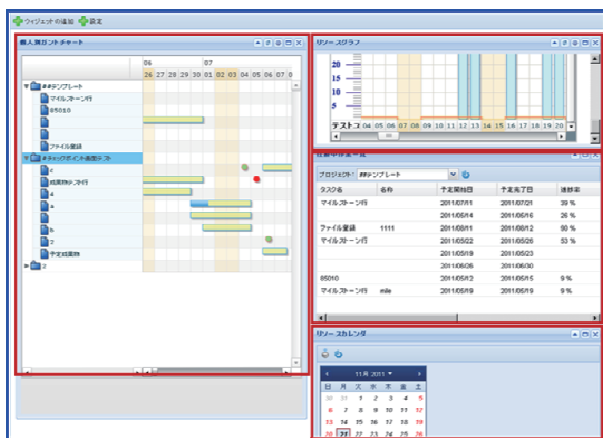


図 5 ダッシュボード画面

ウィジェットの基本仕様とシステムへの登録方法を公開しているため、カスタマイズエンジニアが Web API を使って製品本体とは独立したモジュールとしてカスタムウィジェットを開発することが可能である。カスタムウィジェットは S2 本体が稼働中であっても動的にシステムに追加し、ダッシュボード画面に配置できるようになる。

4. 効果

S2 が Web API とモディフィケーションのないカスタマイズ機能を備えたことによる効果を述べる。

カスタマイズ開発参画の容易化

開発ライセンス契約により、詳細な API リファレンスやカスタマイズガイド、サンプルソースが提供される。Web API とカスタマイズ機能を利用することで、S2 の詳細を知らないカスタマイズエンジニアでもカスタマイズ開発を行えるようになる。

カスタマイズ開発部分の再利用

モディフィケーションのないカスタマイズ機能により、アドオン画面などのカスタマイズ開発部分をカタログ化しておくことができるようになる。さらに、API の後方互換性を確保することにより、将来製品バージョンが上がってもアドオン部分には手を加えることなくそのまま再利用できる。

ノウハウを効率的に蓄積・共有するための仕掛けとして、ビジネスパートナー向けの拡販情報ポータルサイトを日立 TO で運営する構想がある。

拡販活動へのサポート

カタログ化された過去のカスタマイズ開発部分を活用

するにより、販社であってもユーザに対し説得力のある営業活動が可能になる。他社の過去事例の紹介とともに実際に動作するアドオン機能を使用してデモすることができ、さらに訴求力が高まる。

ユーザの視点では、他社で実績のあるオプション機能が安く早く使えるようになることは大きな魅力となる。

利用形態に合わせたシステムの実現

Web API による柔軟なデータ連携やアドオン機能を利用した画面でのシステム間連携が実現でき、ユーザにとって使いやすく価値のあるソリューションが実現できる。

カスタム属性によるエンドユーザレベルのカスタマイズも利用を促進する要因となる。

開発力のあるユーザであれば、開発ライセンスを契約いただくことで、Web API を利用した独自のエクセルマクロやスクリプト言語を利用したツールを追加開発することも可能である。

オンデマンドな機能追加

運用開始後に状況が変化し新しい課題が持ち上がった場合でも、必要になった段階でオプション機能の中から課題にマッチするものを選ぶ、もしくは新たにカスタマイズ開発を行うことで、タイムリーに機能追加することが可能である。

5. 今後のビジネス展開

ここまで述べてきたように、S2 は研究試作を通して、後方互換性が維持される API とモディフィケーションのないカスタマイズ機能を備えることが確認できた。S2 は製品化され、2013 年 1 月にバージョン 1.0 が発売される予定である。これにより、日立 TO 以外のビジネスパートナーによるソリューション構築や拡販活動が可能になる。

今後のビジネス展開としては、早期に第 3 段階に進めるよう、日立グループや他のベンダとの連携を強化し、「SynViz 協業ビジネス」の実現を目指す。

具体的には、ビジネスパートナー向けの情報提供ポータルサイトを構築し、アドオン導入事例や拡販ツール、デモデータなどのコンテンツ共有や、開発者向けの教育プログラム整備などを予定している。

すでにいくつかのパートナーと具体的な協業の取り組みを始めており、ここで代表的な 2 例を紹介する。

5.1 プラント EAM 連携の取り組み

(株)日立製作所 インフラシステム社では、原子力や火力発電所など電力プラントの設備保全を目的とした Enterprise Asset Management (EAM, 設備資産管理) パッケージ製品 (以下、プラント EAM) を開発している。

プラント EAM は保全計画管理, 作業管理, 構成管理, 設備台帳, 保全文書管理などの機能を持つが, その中で工程管理での計画情報の可視化・編集機能を S2 および SynViz/MP と連携して実現する構想がある。(図 6)

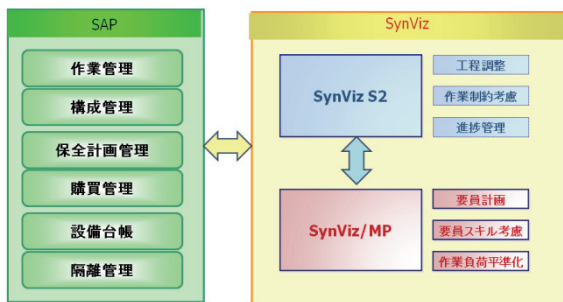


図 6 プラント EAM の製品機能関連図

現在, S2 のプロトタイプを利用して SAP とのシステム連携の評価を行っている。保全計画管理, 作業管理などを担当する SAP から作業計画を送信し, S2 に取り込んでガントチャートに表示し, S2 の画面で計画調整を行い, 結果を SAP に送り返すという一連の業務の流れに沿った連携処理を評価している。

ここで得られた評価結果は, 今後の S2 側の製品改善にも利用される。

プラント EAM が目指している, 多くの機能がシームレスに統合されたソリューションの実現のために, S2 の柔軟なカスタマイズ機能やデータ連携機能が十分に活用できるものとする。

5.2 SmartFAM 連携の取り組み

SmartFAM は, (株)日立情報制御ソリューションズが販売する一般企業向けの設備管理システムである。

S2 と連携してガントチャートを活用した日程・工程データの可視化およびグラフィカルな調整インターフェースとして利用する構想がある。

S2 に加えて, CoreExplorer による故障データの自由記述を含めた原因分析や, Cognos による保全コストの定量分析など, 複数の製品を組み合わせ「連携ソリュ

ーション」を実現することで, ユーザにとっての価値がさらに高まるものと期待される。

S2 が目指す協業ビジネスモデルの成果となるよう推進していく予定である。

6. おわりに

業種・業務向けのニーズにマッチしたソリューションを提供するという SynViz のコンセプト実現のため, 新しい SynViz 製品基盤となる S2 を製品化した。S2 は REST アーキテクチャを採用した Web API およびモディフィケーションのないカスタマイズ機能を備える。

それによって, 業種・業務固有のカスタマイズ機能がノウハウとして蓄積と再利用可能な素地ができた。

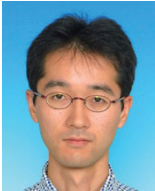
今後は S2 と蓄積したノウハウを活用し, 他社製品と組み合わせた連携ソリューションビジネスを推進していく。そのため日立グループ会社を始めとする, ビジネスパートナーとの協業関係を強化していく。

参考文献

- 1) R. Fielding: Architectural Styles and the Design of Network-based Software Architectures, Chapter 5 Representational State Transfer, 2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, (2012 年 9 月閲覧)
- 2) Amazon.com, Inc.: Amazon Web サービス, <http://www.amazon.co.jp/gp/feature.html?docId=451209>, (2012 年 9 月閲覧)
- 3) Yahoo Japan Corporation: WebAPI の使い方 (GET リクエスト), <http://developer.yahoo.co.jp/appendix/request/rest/get.html>, (2012 年 9 月閲覧)
- 4) D. Crockford: The application/json Media Type for JavaScript Object Notation (JSON), RFC 4627, July 2006.
- 5) Martin Gudgin, et al.: SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), The W3C XML Protocol Working Group, 27 April 2007, <http://www.w3.org/TR/soap12-part1/>, (2012 年 9 月閲覧)



庄子 真海 1999 年入社
パッケージ開発部
S2 プロダクトマネージャ
shojim@hitachi-to.co.jp



門司 太郎 1994 年入社
研究開発部
S2 アーキテクチャ設計
monji@hitachi-to.co.jp



竹本 宗弘 2000 年入社
パッケージ開発部
S2 設計開発
takemoto@hitachi-to.co.jp



内海 宏律 2006 年入社
研究開発部
自社パッケージソフトウェア, コンポ
ーネントの設計, 試作, 開発
hironori.utsumi.01@hitachi-to.co.jp