

次世代統一 AP 基盤 CoreCenter BASE の役割

Role of CoreCenter BASE; Next Generation Standardized Application Infrastructure

佐々木 啓 将

要 約 CoreCenter[®] BASE は流通業向けの各業務アプリケーションで利用されることを想定し、標準化や生産性/品質向上、開発工期の短縮の実現を目標として開発されたシステム基盤ソリューションである。システム基盤として実行基盤だけでなく、開発基盤としての役割を持ち、且つ流通業において広く使われるであろうユーザ管理などの業務支援機能を備えている。

今般、適用事例が増えると共に様々な機能要求も蓄積されてきており、更なる機能拡充を計画中である。

Abstract CoreCenter[®] BASE is assumed to be used as business application for distribution industry and is system infrastructure solution developed with the goal of realizing standardization, improving productivity / quality and shortening the development term of works. As a system infrastructure solution, CoreCenter BASE has the role not only as practice base but as development base and equipped with business supporting functions such as user management etc. that are widely used in distribution industry.

In addition, we are planning further functional expansion for this solution as functional requirements have been accumulated due to the increase of application examples.

1. は じ め に

CoreCenter[®] BASE は日本ユニシスが2011年7月より提供している流通業態向けの基盤ソリューションである。複数の業態向けソリューションの基盤機能として開発されたソリューションであるがその適用範囲は広く、個別のアプリケーション開発時の基盤ソリューションとしても適用が可能である。

Microsoft[®] Windows[®] と .NET フレームワークを採用し、Web アプリケーションの実装をターゲットとしたソリューションであり、CoreCenter BASE を利用することで、開発や導入作業の生産性向上と開発標準化の推進が可能である。

本稿では、2章でCoreCenter BASE を構築した背景と目的並びにその目的を達成するために採用したアーキテクチャ構造を記述し、3章でCoreCenter BASE が持っている各機能を記述する。

2. CoreCenter BASE の位置付けと構造

2.1 CoreCenter BASE 構築の背景と目的

CoreCenter BASE は、複数の流通業向けソリューションの共通基盤としての役割を担うべく開発されたソリューションである。その背景としてはシステム開発において、品質/生産性/保守性向上が求められる中、開発期間の短縮化傾向や開発を担当した要員の固定化を防ぐ（手離れをよくする）ニーズや標準化された基盤のニーズがあった。開発には各業態担当者が集ま

り市場や現場からの要求を踏まえソリューションの方針を検討し、以下の目的を定めた。

- ・開発成果物の再利用がし易いアーキテクチャ構造の採用
- ・ビジネスロジックの入れ替えを容易に実現できるアーキテクチャ構造とする開発部品や、実行時ユーティリティを準備することにより開発負荷の低減を図る
- ・ビジネスロジック機能をパラメータ設定のみで調整できるアーキテクチャ構造とする導入/構築作業の標準化、並びにテンプレートを準備する
- ・共通基盤として多方面の業態システムに採用されることで、業務知識の属人化による要員の固定化を防ぐ

2.2 CoreCenter BASE のねらい

CoreCenter シリーズはアーキテクチャの標準化を図り、品質/生産性/保守性の向上や開発期間の短縮を目的として開発されたソリューション群である。CoreCenter BASE は CoreCenter シリーズの基盤ソリューションとして、目的実現のためのフレームワークや部品群を提供しているが、各業態ソリューションとは機能を分離した形で開発しており、特定の業務アプリケーションに依存せずに適用が可能である。

このように、CoreCenter BASE 適用の裾野は広く、新たなソリューションの基盤システムとしての活用や、個別スクラッチ開発案件への適用等も十分に可能で、それに向けての機能拡充も視野に入れている。

2.3 CoreCenter BASE のアーキテクチャ

CoreCenter BASE の企画段階では、広く多種多様な業態のシステムに適用できるよう、特定業態固有の要素は極力排除し、実装する機能は汎用性/独立性を保つべく検討した。

2.3.1 CoreCenter BASE の構造

CoreCenter BASE の全体構造を図1に示す。実装技術のアーキテクチャとしては、市場シェアやニーズから Windows ベースのアーキテクチャを採用しており、データベースエンジンは SQLServer を前提としている。さらに日本ユニシスのオープン系システム基盤である Atlas-BASE^{*1} の上で動作する構造となっている。具体的にはアプリケーション開発標準ツールである MIDMOST for .NET Maris を採用し、実行フレームワークや開発標準ルールの提供を実現している。また、実装環境としてはオンプレミスだけでなく、クラウド環境もサポートする等、多様な環境での稼働に対応している。

これらのアーキテクチャ上で、CoreCenter BASE は大きく三つの機能・サービスを提供している。

一つめはフレームワーク層としてトランザクション管理や例外処理等実装システムを動作させるための実行ユーティリティ群及び、画面ジェネレータや予め用意された画面コントロール部品等の UI (UserInterface) コントロールを提供している。

二つめは共通機能として、殆どの業務アプリケーションにおいて非機能要件として必要となるユーザ管理やバッチ実行制御機能等を提供している。

三つめは開発 suite として、品質/開発生産性向上や標準化のためのガイドや規約のドキュメントと設計書テンプレートを提供している。

また、CoreCenter BASE は、外部連携の EDI 用ソフトウェア、帳票出力用ソフトウェア、システム監視用ソフトウェア等、いくつかのサードベンダ製品を採用することで、多種多様な機能を提供している。

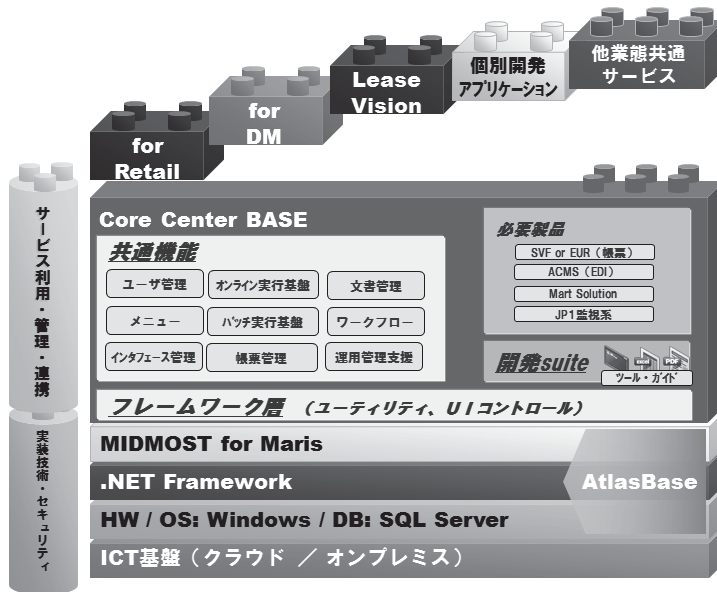


図1 CoreCenter BASE 全体構造

2.3.2 SOA の採用

CoreCenter はアーキテクチャに SOA^{*2}を採用している。これは CoreCenter BASE の機能だけでなく、CoreCenter BASE 上で稼働する業態向けアプリケーションにおいても、入出力定義を標準化して各サービスを疎結合連携とすることでコンポーネント化を図り、利用者が必要なサービスを必要に応じて利用可能とするものである。また、各サービス内でもパラメータ設定のみで機能を選択可能とする構造をとっており、導入先毎の個別カスタマイズを極力減らす考慮がなされている。図2は小売を例としたもので、CoreCenter BASE は各サービス間の連携基盤として機能を提供している。

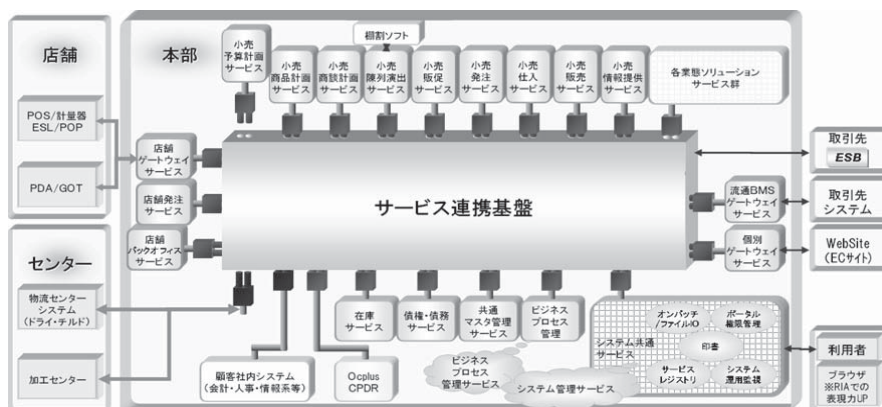


図2 SOA イメージ

2.3.3 アプリケーション構造

次に SOA を実現するための CoreCenter BASE のアプリケーション構造について説明する。CoreCenter BASE は図 3 に示す 4 階層の論理構造（以下レイヤーと表記）をとっており、DI^{*3}（Dependency Injection）の考えに則ったアーキテクチャ構造として、各レイヤー間の関連付けもプログラムから独立させ、依存性を排除している。具体的には、各レイヤーの役割を明確に定め、実装機能範囲を区別しレイヤー間の依存関係を小さくすることで、あるレイヤーの実装の変更が及ぼす他のレイヤーへの影響を最小限に抑え、保守性を向上している。以下に各層の役割を説明する。

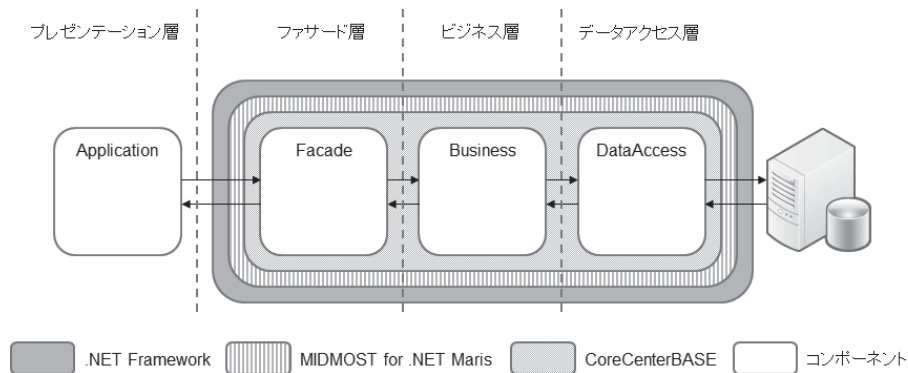


図 3 CoreCenter BASE レイヤー構造

i) プレゼンテーション層の役割

Web アプリケーション（画面）、バッチアプリケーション、Web サービスアプリケーションを持つ。各アプリケーションから CoreCenter BASE で提供されるファサードリストを用いてファサードを呼び出す。ここでは、クラスを動的に実体化しており、呼び出すファサードを変更したい場合は、このリストを修正することで、プログラムを変更することなく実装が可能となる。

ii) ファサード層の役割

ファサード層はビジネス層もしくは API 部品の呼び出しを担う。ファサード層からビジネス層を呼び出す場合は CoreCenter BASE で提供されるビジネスリストを利用する。呼び出すビジネスを変更したい場合は、このリストを修正することで、プログラムを変更することなく実装が可能となる。

iii) ビジネス層の役割

業務処理ロジックが実装される場所となる。データベースへのアクセスがある場合はビジネス層からデータアクセス層を呼び出す。

iv) データアクセス層の役割

データベースへのアクセスを担う。

CoreCenter はこのようなアーキテクチャをベースに業務機能の単位でファサード以下の組み合わせをコンポーネント化し、一つのサービスとして提供する構成となる。

3. CoreCenter BASE で提供する機能

本章では CoreCenter BASE で提供する機能について記述する。CoreCenter BASE は、前章でも述べたように、

- ・フレームワーク層の機能（実行ユーティリティ、開発部品）
- ・共通機能（流通業態で広く利用される機能をサブシステム化し実装）
- ・開発 suite（ドキュメント等）

で構成される。3.1 節以降でそれぞれの機能について説明する。

3.1 フレームワーク層の機能

CoreCenter BASE では、実装システムを動作させるためのユーティリティと、アプリケーション開発に利用されるツールや部品を提供している。

3.1.1 ユーティリティ

CoreCenter BASE は実行基盤として、MIDMOST for .NET Maris の機能と連携し、いくつかのユーティリティを提供している。各業務アプリケーションの開発者は開発規約に則って設計・開発することで個別のコーディングは不要となり、効率的なシステム開発が可能となる。また、合わせてユーティリティが担う機能の標準化も図れる。主要なユーティリティ機能について以下 1) から 5) で説明する。

1) トランザクション管理

トランザクションの管理機能は CoreCenter BASE のベースクラスが担っており、規約やテンプレートに則って実装することで開発者は意識せずにファサード単位での管理を実現している。これによりビジネスロジックからトランザクションを独立させ、ビジネスロジックの変更によるトランザクションスコープの変化を防ぎ、ビジネスロジックの変化対応力を担保している。CoreCenter BASE でのトランザクションイメージを図 4 に示す。

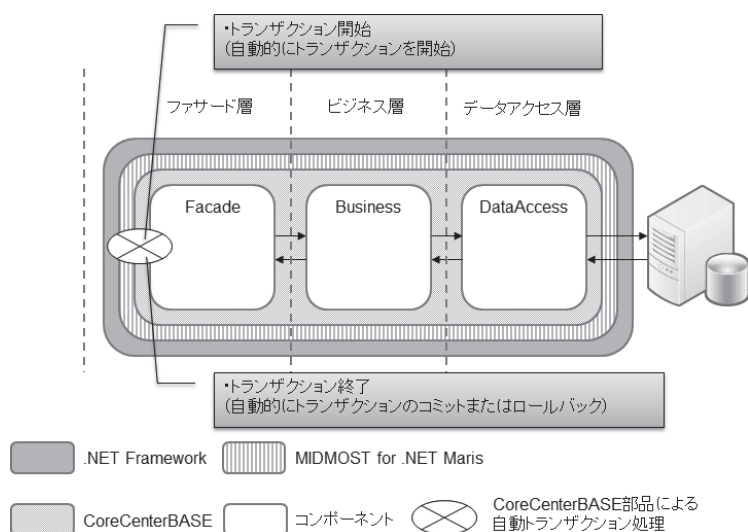


図 4 トランザクション管理

2) 例外処理

CoreCenter BASE は例外処理を管理する機能を持っているので、業務アプリケーションの開発者は開発規約に則ることでコーディングを意識せず例外捕捉の機能が実装可能となる。捕捉された例外は、例外の発生場所に応じて、自動的にそれぞれのログへ出力される。

3) 入力値検証

CoreCenter BASE では入力値検証のために、単項目チェックの機能を備えている。単項目チェックとは、ある項目の入力値が、その項目が持つ意味合いに対して妥当な値かを検証することである。例えば、“商品コードとして入力された値は、4桁の数字である”等を検証する。CoreCenter BASE では、単項目チェックルールについて、その内容ごとに CheckID を振り、システム内で一元管理する仕組みを提供している。

単項目チェックのタイミングは大きく分けて以下の二つである。

- i) 各アプリケーション形態の入り口のタイミングにて入力値検証違反を業務処理の前に検知する目的で行う（例：画面の入力項目等）。
- ii) ビジネス層におけるデータ更新前のタイミングでデータ保護の目的で行う。これはビジネス層にたどりつく前に入力値検証がしっかり行われていれば必要のない処理だが、ビジネス層は、どこから呼ばれるかを選べないので、ビジネス層単体でも整合性が保てるように構築するためである。

これらを実装するために CoreCenter BASE では、Web 画面向けに「Web 検証コントロール」、バッチ/Web サービス/ビジネス層での実装向けに「単項目チェック API」、データセット経由でのデータベース更新向けに「テーブルバリデーター」を提供している。

4) 閉塞処理

システム運用上、機能毎に利用時間を設ける場合が考えられる。そのために閉塞管理機能を提供している。CoreCenter BASE のユーザ管理機能にて閉塞時間帯の情報を保持し、CoreCenter BASE が提供するベースクラスにてチェックすることで、業務アプリケーションはそれを意識することなく開発が可能となる。閉塞の種類として予閉塞と本閉塞があり、それぞれ別の時間帯設定が可能である。

予閉塞とは、本閉塞前に新たな利用者を増やさないことを目的とし、新たなセッションの確立は制限し、既存のセッションのみ可能とさせる機能である。既に接続している利用者はそのまま利用は可能だが、新たな利用は制限されることになる。それに対し本閉塞とは既存のセッションも閉塞時間到来以降はセッションの維持ができなくなる。

5) 多重度管理

業務システムによっては、同時に実行させたくないアプリケーションが存在するケースがある。そのために同時実行数を制御する多重度管理の機能を提供している。これはある処理に対して、処理を実行できるユーザの数を制限したり、複数のユーザが同時に実行できないようにするための機能であり、CoreCenter BASE のファサードベースクラスにて実装している。

3.1.2 開発部品

CoreCenter BASE では開発者向けの支援部品群を提供している。大きくは開発作業における手順を効率化するためのツールと、開発作業量を減らすために組込部品として提供される部

品群とに分けられる。以下に開発作業手順効率化のための画面ジェネレータと、組込部品として提供している画面コントロール群について記述する。

1) 画面ジェネレータ

画面ジェネレータは CoreCenter BASE 提供の画面物理設計書からの情報を基に以下 i) から iv) の四つのファイルを自動生成するツールである。

表 1 画面コントロール一覧

機能名	機能概要
標準ボタン	ASP.NET のコントロールを拡張したもの。
標準チェックボックス	ASP.NET のコントロールを拡張したもの。
標準チェックボックスリスト	ASP.NET のコントロールを拡張したもの。
標準ドロップダウンリスト	ASP.NET のコントロールを拡張したもの。 Value と Text の値を「:」で連結して、レンダリング時に「Value : Text」という表示ができるようになるプロパティが拡張されている。
標準ファイルアップロード	ASP.NET のコントロールを拡張したもの。
標準テキストボックス	ASP.NET のコントロールを拡張したもの。 IME モードの指定が可能。(HTML の仕様に従って)
標準ハイパーリンク	ASP.NET のコントロールを拡張したもの。 新しいウィンドウを開く際に、サイズを指定したり、メニューバーやステータスバーを消したりするオプションを指定することができるように拡張している。
標準イメージ	ASP.NET のコントロールを拡張したもの。
標準イメージボタン	ASP.NET のコントロールを拡張したもの。
標準イメージマップ	ASP.NET のコントロールを拡張したもの。
標準ラベル	ASP.NET のコントロールを拡張したもの。
標準リンクボタン	ASP.NET のコントロールを拡張したもの。
標準リストボックス	ASP.NET のコントロールを拡張したもの。
標準リテラル	ASP.NET のコントロールを拡張したもの。
標準パネル	ASP.NET のコントロールを拡張したもの。
標準ラジオボタン	ASP.NET のコントロールを拡張したもの。 グリッドの内部で利用された時に指定できるグループを追加している。
標準ラジオボタンリスト	ASP.NET のコントロールを拡張したもの。
標準ツリービュー	ASP.NET のコントロールを拡張したもの。
標準隠しフィールド	ASP.NET のコントロールを拡張したもの。
標準チャート	ASP.NET のコントロールを拡張したもの。
モーダルボタン	ASP.NET のボタンコントロールを拡張したもの。 モーダル画面が閉じられた時に、ポストバックをするかどうかを設定できる。
モーダルリンクボタン	ASP.NET のリンクボタンコントロールを拡張したもの。 モーダル画面が閉じられた時に、ポストバックをするかどうかを設定できる。
ロストコンプリートテキストボックス	ロストフォーカス時、入力値を引数として指定した Web サービスを呼び出し、結果を対象コントロールの Text プロパティに設定する TextBox。
日付入力 TextBox	日付入力専用の TextBox。
タブ	タブ付きパネルを生成する。
一覧 GridView	Infragistics 社の WebDataGrid を継承したコントロール
キャプションラベル Label	項目のキャプションを表示する Label。
必須ラベル Label	必須項目であることを示す半角アスタリスクを表示する Label。
検証実行 (画面 UI バリデーション コントロール)	検証内容定義クラスに基づき、指定した項目に対して検証を行う。 検証結果が不可である場合、不可であることを示すため、当該コントロールの背景色は赤になる。
二重送信防止	・ポストバック/非同期ポストバック(マスターページのポーリングによるものを除く)の発生時にインジケータが表示され、画面上のコントロールをクリックできなくなり、Tab キー押下でもコントロールにフォーカスが当たることを制御している。

- i) 画面の大まかなレイアウトを定義する *.aspx ファイル
- ii) 画面の操作に対する振る舞いを記述する *.aspx.cs ファイル
- iii) 画面に配置するコントロールの一覧を定義する *.aspx.designer.cs ファイル
- iv) 画面レイアウトを HTML 形式で記述した *.html ファイル

本ツールはこれらのファイルを新規に作成する手間から開発者を開放し、作業の効率化を図る目的で用意されている。

2) CoreCenter BASE で提供するコントロール部品

CoreCenter BASE では、ASP.NET やサードベンダの製品等をベースに拡張機能を持たせた画面開発用の画面コントロール群を提供している（表 1）。これらのコントロールを利用する場合は CoreCenter BASE がテンプレートとして提供する物理設計書を用い、利用するコントロールを選択し、画面ジェネレータ機能を利用することで、コーディングレスで画面プログラムを作成することができる。

3.2 共通機能

CoreCenter BASE では、各業態アプリケーションに共通して必要な機能を共通機能として提供している。複数のサブシステムで構成されており、画面機能/バッチ機能、業務アプリケーションから各サブシステムとインタフェースするための API を提供している。業務アプリケーションは共通機能の範囲については個別に開発する必要はなくなり、アプリケーション全体の標準化に寄与する。共通機能を構成するサブシステムを以下に説明し、詳細を表 2 に挙げる。

・ユーザ管理

ユーザ/パスワードによるログイン管理を実装する。所属情報と権限情報も合わせて保持し、ログインユーザのシステム上の振る舞いをコントロールする。

・メニュー

ログイン画面並びにメニュー画面の機能を持つ。ユーザ管理と連携し、ログインしたユーザの権限に応じたメニューを表示する。

・インタフェース管理

外部とのインタフェースのための管理機能を持つ。外部接続は ACMS^{*4} と連携し各インタフェースプロトコルによる通信を実現する。クエリ登録機能を持ち、簡単なインタフェースであればプログラムレスでの実装が可能である。

・バッチ実行

バッチ処理の管理機能を持つ。ジョブネットやジョブのスケジュール実行及び即時実行が可能である。

・帳票管理

SVF または EUR と連携した帳票管理の機能を持つ。帳票毎のクエリ登録機能を持ち、簡単な帳票であればプログラムレスでの実装が可能である。

・文書管理

ドキュメントの管理機能を持つ。ロールと紐付きユーザ毎の登録/参照が可能である。

・ワークフロー

業務アプリケーションと連携し申請/承認等のワークフローを実装する。ワークフローを制御するサービス機能と、実行中のワークフローの状態を監視/操作する機能を提供する。

表 2 共通機能一覧

サブシステム	機能概要	機能説明
ユーザ管理	ログイン管理	ユーザ/パスワード情報を保持し、システムへのログイン認証を行う。パスワードについては有効期限や設定可能な文字種類等を管理する。
	権限管理	組織/役職情報と連携し、各ユーザに対するロール（権限）の管理を行う。
	機能認可	ロールと連携し利用可能なメニュー/画面/バッチ処理の実行権限等を設定/管理する。
	情報提供	各種ユーザ情報を取得可能な API を提供。この API を利用することで他システムへの情報連携の実装が可能。
メニュー	メニュー画面	メニューテンプレートを準備。メニュー構成を設定シートに記載し、システムへ取り込むことで開発なしでメニューを実装可能。
	お気に入り管理	ユーザ毎に良く使う画面をお気に入りとして登録可能。お気に入りは任意の複数フォルダでの保持が可能。
	パーソナライズ	ユーザ毎にメニューに表示する画面を表示/非表示のコントロールが可能。
インタフェース管理	マスタ管理	他サーバや外部との接続情報（相手先、データ種別、プロトコル）をマスタで一元管理しコントロールする。
	データ作成	送信データ作成のためのクエリを保持。これによりデータ作成のためのプログラムを作ることなく、インタフェースの実装が可能となる。
	ACMS 連携	外部とやりとりをするために、ACMS との連携機能を準備。マスタ情報や送受信指示の ACMS への連携機能を有する。
	処理状況確認	インタフェース処理の確認画面が準備されており、ACMS の画面を立ち上げることなく状況確認が可能。また、再実行等の操作も可能。
	ファイル変換	AnyTran [®] と連携しファイルの変換（コード変換、フォーマット変換）の自動化が可能。事前に変換定義を行うことで処理を自動化可能。
バッチ実行基盤	ジョブネット実行	ジョブ並びにジョブネットを登録することで、バッチ処理の実行管理が可能（スケジュール実行、ファイル着信実行、手動実行）。
	多重度制御	ジョブネット単位で同時実行可能なジョブネット数をコントロール可能。
	処理状況確認	ジョブネット（ジョブ）の実行状況を画面から確認可能。該画面からジョブネット（ジョブ）の再実行操作も可能。
帳票管理	マスタ管理	帳票作成先（PDF、印書、電子帳票連携）や、利用するフォーム情報等をマスタで一元管理する。
	データ作成機能	マスタ上へクエリを保持することで帳票出力時のこのクエリにてデータ作成が可能で、プログラム作成することなく実装が可能となる。
	データ連携機能	帳票データとして、CSV を連携（取込）帳票を出力することが可能。
	仕訳機能	帳票データに含まれる特定の項目を指定することで、指定項目のキーブレイクを条件として帳票の仕訳が可能。
	処理状況確認	帳票出力状況を画面から確認することが可能。該画面から再印書の指示も可能。
文書管理	文書登録機能	任意の文書（PDF、CSV、他）を特定ロール向けに登録することで、該当ロール保持者へ文書を展開する機能。
	文書閲覧機能	個人向けに登録された文書を確認する画面機能。
ワークフロー	プロセスコントロール	ワークフローのプロセスを登録することで、プロセスのコントロールを行うことが可能。
	業務 AP 連携	ワークフローの状態遷移を制御するための API を提供。ワークフロー上利用される業務 AP と連携することでワークフローを実装する。
	状況確認画面	実行されているワークフロープロセスの状況確認、制御画面を準備。
運用管理支援	バックアップ機能	DB やファイル等のバックアップ部品を準備。
	ログ機能	システムから出力される各種ログを DB へ取り込む機能を準備。
	ログ参照機能	DB へ取り込まれたログ情報を参照する画面を準備。サーバ接続し直接テキストログへアクセスすることなくログ調査が可能。
	ガベージ機能	DB テーブルや、ファイルのガベージ機能を準備。ガベージ条件等の設定を登録するのみでガベージ機能の実装が可能。
	リストア機能	バックアップした情報からデータをリストアする機能。

- ・運用管理支援

データベースやファイルのバックアップ/ガベージ/リストア機能を持つ。また、各種ログの収集と参照機能を提供する。

3.3 開発 suite

CoreCenter BASE では開発 suite として開発者向けの各種ドキュメント類を提供している。開発時にこれらのドキュメントを利用することで、生産性向上並びに開発標準化を図ることを目的としている。表3に主なドキュメントについて記述する。

表3 CoreCenter BASE ドキュメント情報

ドキュメント種類	ドキュメント数	機能概要
論理設計書テンプレート	11	論理設計に利用する標準の設計書テンプレート。
物理設計書テンプレート	24	物理設計にて利用する標準の設計書テンプレート。Excel マクロにより一部設計情報の実装環境への引き継ぎが可能。
開発規約	15	開発時に守るべき規約類。CoreCenter BASE の機能を利用するためのルールを記載。
ガイド	42	開発者向けや運用者向けなど、CoreCenter BASE を利用するための各種ガイドを記載。

開発者が上記のテンプレートを利用し、開発規約とガイドに従って開発することで、CoreCenter BASE が提供する機能を特に意識することなく実装することが可能となる。

4. CoreCenter BASE の今後

CoreCenter BASE はリリースされ間もなく2年半が経とうとしている。時間の経過と共に、ユーザ数の増加や当初想定していなかった業態への適用も進んでいる。今後更に多方面の顧客へ適用し、顧客の価値向上に貢献するために、以下の対応を予定している。

- ・機能の拡充

適用事例の増加や、新たな業態への適用等が進み、機能の拡充や新機能の追加要求が出ており、汎用的な機能について機能拡張を進める予定である。

- ・最新バージョンの対応

OS や SQLServer 等、CoreCenter シリーズで扱うソリューション製品は多く、それらの最新バージョンへの対応を継続して実施する予定である。

- ・多ブラウザへの対応

今後、一般利用者向けのアプリケーションの CoreCenter BASE 上での実装も考えられる。一般利用者向けの場合には、ブラウザの種類を限定することは難しく、CoreCenter BASE としても多ブラウザのサポートが必要となる可能性があるため、一般利用者向けシステムのニーズや、ブラウザソフトのシェアを調査しながら多ブラウザ対応を検討していきたい。

5. お わ り に

CoreCenter BASE は業務ソリューションを稼働させるための基盤ソリューションとして誕生したが、その適用の裾野は広く、個別開発にも採用可能なソリューションとなっている。今後、.NET フレームワークにて Web アプリケーションを開発する場合にも CoreCenter BASE が採用されることを期待している。

また、CoreCenter BASE は基盤ソリューションということもあり、カバーする機能範囲や保持している機能の詳細等分かりにくいと思われるが、本書が理解の一助になれば幸いである。

最後に本書を執筆するにあたり、様々な助言/指導をくださった方々に誌面を借りて深く御礼申し上げる。

-
- * 1 AtlasBASE：日本ユニシスグループのシステム基盤領域を対象とした商品・ナレッジサービス
 - * 2 SOA：Service-Oriented Architecture. ソフトウェアを部品化することでサービス共有化を図る手法
 - * 3 DI：Dependency Injection. コンポーネント間の依存関係をソースコードから取り除くことでプログラム間での依存関係を持たないようにしている
 - * 4 ACMS：データアプリケーション社の企業間連携用ソフトウェア
 - * 5 AnyTran：データアプリケーション社の ETL ソフトウェア

執筆者紹介 佐々木 啓 将 (Hiroyuki Sasaki)

1992 年日本ユニシス(株)入社。A/NX シリーズユーザー向け開発業務を担当。その後流通小売業のサービスや開発業務に携わり、2011 年から CoreCenter BASE 主管業務に従事。

