

SEWB+ 概説

概説書

3020-3-B80

対象製品

P-2451-1444 SEWB+ 基本開発環境セット 04-00 (適用 OS : Windows XP , Windows Vista , Windows 7 , Windows Server 2003 , Windows Server 2003 x64 , Windows Server 2003 R2 , Windows Server 2003 R2 x64 , Windows Server 2008 , Windows Server 2008 x64 , Windows Server 2008 R2)

注 WOW64 (Windows On Windows 64) 環境だけで使用できます。

この製品は、ISO9001 および TickIT の認証を受けた品質マネジメントシステムで開発されました。

輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

ERwin は、Computer Associates International,Inc. の登録商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft Excel は、米国 Microsoft Corporation の商品名称です。

Microsoft Office Excel は、米国 Microsoft Corporation の商品名称です。

Microsoft Office Word は、米国 Microsoft Corporation の商品名称です。

Microsoft Word は、米国 Microsoft Corporation の商品名称です。

ODBC は、米国 Microsoft Corporation が提唱するデータベースアクセス機構です。

Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

Visual Basic は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Visual C++ は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Xupper は、ケン・システムコンサルティング (株) の商標です。

その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

表記		正式名称
Microsoft Excel		Microsoft(R) Excel
		Microsoft(R) Office Excel
Microsoft Word		Microsoft(R) Word
		Microsoft(R) Office Word
Visual Basic		Microsoft(R) Visual Basic(R)
Visual C++		Microsoft(R) Visual C++(R)
Windows	Windows XP	Microsoft(R) Windows(R) XP Home Edition Operating System
		Microsoft(R) Windows(R) XP Professional Operating System
Windows Vista		Microsoft(R) Windows Vista(R) Business 日本語版 (32 ビット版)
		Microsoft(R) Windows Vista(R) Enterprise 日本語版 (32 ビット版)
		Microsoft(R) Windows Vista(R) Ultimate 日本語版 (32 ビット版)
		Microsoft(R) Windows Vista(R) Business 日本語版 (64 ビット版)
		Microsoft(R) Windows Vista(R) Enterprise 日本語版 (64 ビット版)
		Microsoft(R) Windows Vista(R) Ultimate 日本語版 (64 ビット版)
Windows 7		Microsoft(R) Windows(R) 7 Home Premium 日本語版 (32 ビット版)
		Microsoft(R) Windows(R) 7 Professional 日本語版 (32 ビット版)
		Microsoft(R) Windows(R) 7 Ultimate 日本語版 (32 ビット版)
		Microsoft(R) Windows(R) 7 Home Premium 日本語版 (64 ビット版)
		Microsoft(R) Windows(R) 7 Professional 日本語版 (64 ビット版)
		Microsoft(R) Windows(R) 7 Ultimate 日本語版 (64 ビット版)
Windows Server	Windows Server 2003	Microsoft(R) Windows Server(R) 2003, Standard Edition 日本語版
		Microsoft(R) Windows Server(R) 2003, Enterprise Edition 日本語版
	Windows Server 2003 x64	Microsoft(R) Windows Server(R) 2003, Standard x64 Edition 日本語版
		Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition 日本語版

表記		正式名称
	Windows Server 2003 R2	Microsoft(R) Windows Server(R) 2003 R2, Standard Edition 日本語版
		Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition 日本語版
	Windows Server 2003 R2 x64	Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition 日本語版
		Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition 日本語版
		Microsoft(R) Windows Server(R) 2003 R2, Datacenter x64 Edition 日本語版
	Windows Server 2008	Microsoft(R) Windows Server(R) 2008 Standard 32-bit 日本語版
		Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit 日本語版
	Windows Server 2008 x64	Microsoft(R) Windows Server(R) 2008 Standard 日本語版
		Microsoft(R) Windows Server(R) 2008 Enterprise 日本語版
	Windows Server 2008 R2	Microsoft(R) Windows Server(R) 2008 R2 Standard 日本語版
		Microsoft(R) Windows Server(R) 2008 R2 Enterprise 日本語版
		Microsoft(R) Windows Server(R) 2008 R2 Datacenter 日本語版

注

64 ビット版の Windows です。

発行

2012 年 7 月 3020-3-B80

著作権

All Rights Reserved. Copyright (C) 2012, Hitachi, Ltd.

はじめに

このマニュアルは、SEWB+ 全体の概要、SEWB+ でアプリケーションを開発する場合に利用する機能、および推奨する運用方法について説明したものです。

SEWB+ 基本開発環境セットは、SEWB+ 基本開発環境と Groupmax Object Server から構成されます。SEWB+ 基本開発環境は、次に示す機能から構成されます。

- リポジトリ管理機能 (SEWB+/REPOSITORY)
- リポジトリブラウザ機能 (SEWB+/REPOSITORY-BROWSER またはリポジトリブラウザ)
- プログラム構築支援機能 (SEWB+/CONSTRUCTION)
- レコード設計支援機能 (SEWB+/RECORD DEFINER)

対象読者

このマニュアルは次に示す方を対象としています。

- SEWB+ の導入、運用または SEWB+ を使用してのシステム開発を検討している方
- システム開発全体の設定・管理をする方
- システム開発の各工程での設計を担当する方

読書手順

このマニュアルは、利用目的に合わせて次の箇所をお読みいただくことをお勧めします。

マニュアルを読む目的	記述箇所
SEWB+ の概要について知りたい	1 章
SEWB+ を利用した、TP モニタでのシステム設計について知りたい	2 章
システム設計での DB 設計の作業、および DB 設計で取得したデータ項目を辞書に取り込む手順について知りたい	3 章
データ項目辞書と業務ルール辞書、およびアプリケーションプログラム作成と辞書の関係について知りたい	4 章
ユーザインタフェースの設計および DB・ファイルの設計について知りたい	5 章
共通部品の設計および作成について知りたい	6 章
TP モニタでのサーバ AP の設計について知りたい	7 章
プログラム定義およびソースプログラム生成に必要な設定や手順について知りたい	8 章
SEWB+/CONSTRUCTION で作成したソースプログラムのコンパイル、および単体テストについて知りたい	9 章
統合テストの準備および手順について知りたい	10 章
SEWB+/REPOSITORY での資源管理について知りたい	11 章
ドキュメントを効率良く運用するためのポイントについて知りたい	12 章
辞書を効率良く運用するためのポイントについて知りたい	13 章
分散開発を支援するためのエクスポート・インポートの機能および運用方法について知りたい	14 章
SEWB+ を利用したシステム開発のノウハウについて知りたい	15 章
このマニュアルの参考情報について知りたい	付録 A
このマニュアルで使用する用語について知りたい	付録 B

このマニュアルで使用している記号

このマニュアルで使用する記号を次のように定義します。

記号	意味
[]メニュー []ボタン など	メニュータイトル, メニュー項目, ボタンなどを示します。
[A] - [B]	- の前に示した [A] メニューから [B] を選択することを示します。

このマニュアルで使用する画面図と操作説明について

このマニュアルでは特に断りのないかぎり, 次に示す Windows で表示される画面, メニュー名およびアイコンで操作を説明しています。これら以外の Windows では, 画面, メニュー名およびアイコンが異なることがあります。詳細は, Windows のドキュメントを参照してください。

機能名	Windows
SEWB+/REPOSITORY	Windows Server 2003
SEWB+/REPOSITORY-BROWSER	Windows Vista
SEWB+/CONSTRUCTION	
SEWB+/RECORD DEFINER	

COBOL について

このマニュアルの「9. コンパイルと単体テスト」は, プログラミング言語に COBOL2002 を使用していることを前提に記載しています。COBOL2002 以外で作成されたソースプログラムをコンパイル・テストする場合は, それぞれのコンパイラのマニュアルを参照してください。

マニュアルとヘルプを効果的にお使いいただくために

SEWB+ ではオンラインヘルプ (以降, ヘルプと略します) を提供しています。利用の目的に応じて, マニュアルとヘルプを使い分けることをお勧めします。

マニュアルは, SEWB+ の概要を理解したいときや, 自プロジェクトに適した運用を検討するときにお使いください。

ヘルプは, SEWB+ を実際に操作しながら機能の詳細を知りたいときにお使いください。ダイアログの操作方法やメニューバーの利用方法や, 注意事項を参照できます。

目次

1	SEWB+ とは	1
1.1	SEWB+ の紹介	2
1.1.1	SEWB+ の特長	2
1.1.2	SEWB+ の導入の流れ	4
1.2	システム構成	8
1.2.1	C/S システム	8
1.2.2	Web システム	10
1.2.3	パッチシステム	10
1.2.4	ワークフローシステム	11
1.3	SEWB+ の製品	13
1.3.1	SEWB+ 基本開発環境セット (クライアントサーバ)	13
1.3.2	SEWB+ 開発クライアント向け製品	13
1.3.3	SEWB+ ランタイム製品	14
1.4	システム開発の流れ	15
2	システムの設計	17
2.1	システム設計の概要	18
2.2	サーバ AP とクライアント AP の処理方式の決定	19
2.2.1	要求仕様の確認	19
2.2.2	DB および機能分散処理の設計	19
2.2.3	設計内容の検証	20
2.2.4	基盤ソフトウェア構成・システム処理方式の確定	20
2.3	開発方式の設計	21
2.3.1	開発言語および支援ツールの決定	21
2.3.2	各種基準書の設計	21
2.3.3	開発方式の設計	22
2.4	データ項目と業務ルールの作成・検討	23
2.4.1	前工程の確認	24
2.4.2	リポジトリの確認	24
2.4.3	業務ルールの作成	24
2.5	外部仕様の作成・検討・評価	25
2.5.1	画面遷移図の作成	26
2.5.2	代表シナリオの決定	26
2.5.3	プロトタイプの開発方法および評価項目の設定	26
2.5.4	画面・帳票レイアウトの確認	26
2.5.5	再利用できるテンプレートや部品の作成	26

3	DB 設計・データ項目の取り込み	27
3.1	データ分析とデータ設計	28
3.1.1	データモデリング	28
3.2	データ項目の取り込み	29
4	辞書の作成	31
4.1	データ項目辞書	32
4.1.1	単項目（基本項目）	32
4.1.2	継承項目	33
4.1.3	結合項目（レコード項目）	33
4.1.4	データ項目の取り込み方法	34
4.2	業務ルール辞書	35
4.2.1	業務ルールの定義項目	35
4.2.2	業務ルールの登録	36
4.3	AP 設計と辞書	38
4.4	辞書使用時の注意事項	39
5	入出力の設計	41
5.1	ユーザインタフェースの設計	42
5.1.1	GUI 構築ツールでのクライアント AP の設計	42
5.1.2	帳票レイアウト設計	45
5.2	ファイル・DB の設計	46
5.2.1	レコード定義	46
5.2.2	ファイル・DB 設計時の注意事項	47
6	適用テンプレート・部品の準備	49
6.1	テンプレートと部品の作成	50
6.1.1	共通部品の設計	51
6.1.2	共通部品の作成	54
7	サーバ AP の設計	57
7.1	TP モニタを使用したサーバ AP の設計	58
7.1.1	TP モニタ使用時のツールの関連	58
7.1.2	サーバ AP 作成の流れ	59
7.2	帳票出力する AP の設計	62

8	プログラムの作成	63
8.1	プログラム定義の準備	64
8.1.1	SEWB+/CONSTRUCTION での環境設定	64
8.1.2	パス設定	64
8.2	プログラムの作成 (TP モニタ)	67
8.2.1	TP モニタで動作する AP の作成	67
8.2.2	プログラム作成時の注意事項 (TP モニタ)	68
8.3	ソースプログラムの生成と修正手順	69
8.3.1	ソースプログラムを新規に生成する場合	69
8.3.2	修正時の手順	69
8.3.3	生成時および修正時の注意事項	69
9	コンパイルと単体テスト	71
9.1	コンパイルについて	72
9.1.1	ソースプログラムのリポジトリへの登録	72
9.1.2	プログラムのコンパイルとテスト	72
9.1.3	プログラムの修正	72
9.2	COBOL2002 でのコンパイル	74
9.2.1	COBOL 開発マネージャの利用	74
9.2.2	COBOL 開発マネージャの注意事項	76
10	統合テスト	79
10.1	統合テストの準備	80
10.1.1	転送を考慮したドキュメントフォルダ構成	80
10.2	ターゲットシステムでのテストと修正	83
10.2.1	ターゲットシステムが UNIX の場合	83
10.2.2	ターゲットシステムがホストの場合	85
11	資源の管理	87
11.1	リポジトリの機能	88
11.1.1	フォルダでのドキュメント管理と辞書管理	88
11.1.2	ドキュメントの管理	88
11.1.3	ドキュメントの更新	89
11.1.4	ドキュメントの参照	89
11.1.5	関連の設定	89
11.1.6	アクセス権の設定	90
11.2	リポジトリの準備	93
11.2.1	サーバ側およびクライアント側の環境設定	93
11.3	リポジトリでの命名時の注意事項	96

12	ドキュメントの運用	97
12.1	環境設定	98
12.2	ドキュメントフォルダの作成と運用	99
12.3	セキュリティ管理	100
12.4	ドキュメント作成と修正の手順	102
13	辞書の運用	105
13.1	環境設定	106
13.1.1	必要な環境設定	106
13.2	辞書フォルダの分割作成と構成	107
13.2.1	辞書分割時の注意事項	107
13.2.2	データ項目の分類方法	107
13.2.3	分類方法と辞書フォルダ構成	108
13.3	セキュリティ管理	110
13.3.1	辞書のアクセス権の設定	110
13.3.2	排他制御	111
14	分散開発環境	113
14.1	分散開発環境とは	114
14.1.1	配布の形態	114
14.2	資源の配布	117
14.2.1	配布の対象	117
14.2.2	配布の目的	117
14.2.3	配布時の増分管理	117
14.2.4	環境構築情報のエクスポート	117
14.3	エクスポート・インポート時の設定	119
14.3.1	エクスポート時の設定	119
14.3.2	インポート時の設定	123
14.3.3	エクスポート・インポート運用の設定例	124
14.4	エクスポート・インポート時の注意事項	131
14.5	分散運用の流れ	132
15	SEWB+ のノウハウ	133
15.1	開発ライフサイクルとリポジトリ管理	134
15.1.1	開発ライフサイクル	134
15.1.2	UNIX 環境でのシステム統合・実機テスト時の作業の流れ	135
15.1.3	リポジトリへの資源の反映	136
15.2	エクスポート・インポートの自動運用	138

付録	139
付録 A このマニュアルの参考情報	140
付録 A.1 関連マニュアル	140
付録 A.2 このマニュアルでの表記	142
付録 A.3 KB (キロバイト) などの単位表記について	142
付録 B 用語解説	143
索引	149

1

SEWB+ とは

SEWB+ は、基幹システムに要求される高品質、高生産性を実現するための機能を提供する CASE ツールです。この章では、基幹システムのニーズを満たすために SEWB+ が提供している機能の概要を紹介します。

1.1 SEWB+ の紹介

1.2 システム構成

1.3 SEWB+ の製品

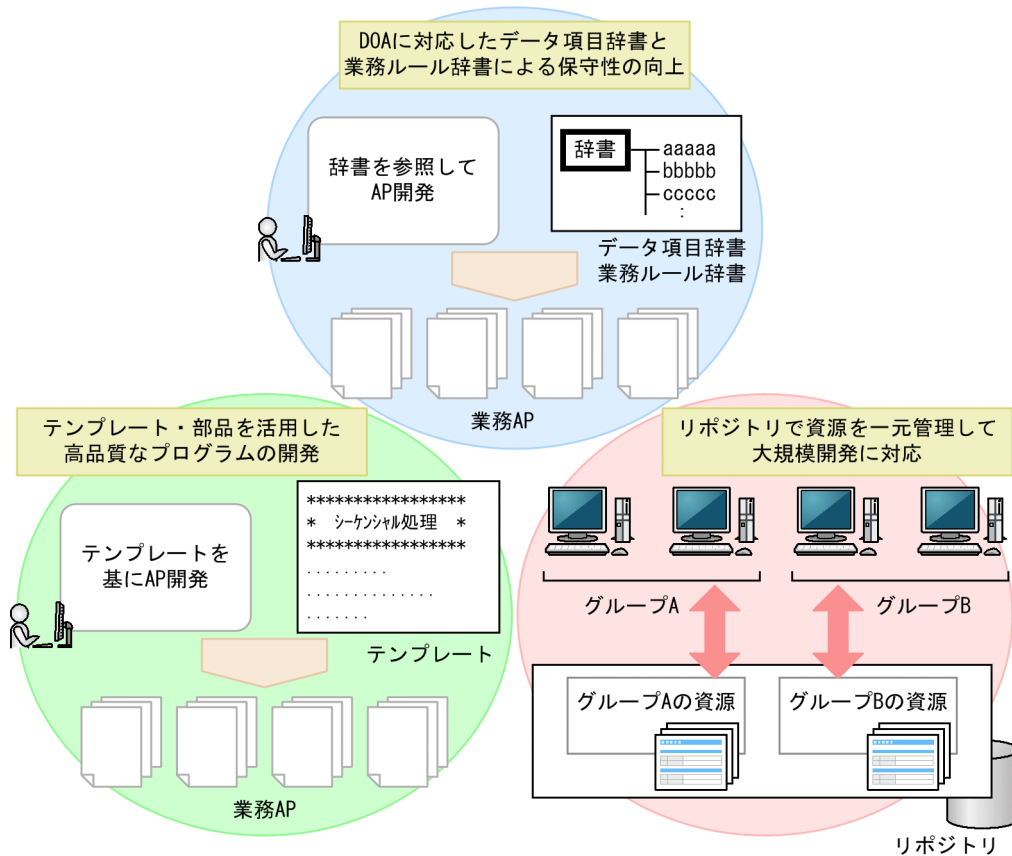
1.4 システム開発の流れ

1.1 SEWB+ の紹介

SEWB+ は、基幹システムに求められる「高品質」、「高生産性」および「大規模開発への対応」を実現させることを目的として開発された製品です。

SEWB+ の概要を次の図に示します。

図 1-1 SEWB+ の概要



1.1.1 SEWB+ の特長

基幹業務の AP 開発を支援するための、SEWB+ の特長について説明します。業務 AP には、次の特性があります。

- 扱うデータの種類、および量が膨大である
- プログラムの中には構造が似ているものが多い
- 大規模開発では、仕様変更や環境変化への柔軟な対応が必要

これらは業務 AP の特性であると同時に、作成する AP の品質を一律に高いレベルに保ったり、レベルを保ったまま生産性を上げたりすることの妨げでもありました。この問題に対処するための、SEWB+ の機能を紹介します。

(1) DOA (データ中心アプローチ) 手法に基づくデータの管理・運用を実現 - 辞書

SEWB+ では「扱うデータの種類、および量が膨大である」という問題に対処するために、辞書を利用します。辞書には、システム開発で扱うデータ項目を登録するデータ項目辞書と、データ項目に着目して、

データ項目特有の処理を部品として登録する業務ルール辞書があります。SEWB+ では、これらのデータ項目や業務ルールを DOA 手法に従って定義・管理し、運用するので、データ同士の整合性が保証されます。

また、SEWB+ の辞書は、ERwin/ERX などのデータ分析ツールや DB 設計ツールが生成したデータを取り込みます。

なお、辞書はシステム開発の資源を統合的に管理するリポジトリの一部です。

(2) 高品質の維持と高生産を実現 - テンプレート・部品

SEWB+ では「プログラムの中には構造が似ているものが多い」というむだを解消するために、テンプレート・部品を利用します。業務 AP で頻出するプログラム構造は定型・抽象化してテンプレートとして整理します。入出力など頻出する処理は部品として整理します。これらを業務ルールと組み合わせることで、高品質で保守性の高い AP を、効率良く作成できます。

(3) 大規模開発に対応する - リポジトリ

開発規模が大きくなると、開発途中の仕様変更の変更波及を解析したり、必要なドキュメントを作成したりするだけでもかなりの工数が掛かります。また、グループ作業をするため、開発資源のバージョン管理やアクセス権の問題が生じます。

(a) 仕様変更などへの柔軟な対応を実現

SEWB+ では「大規模開発では、仕様変更や環境変化への柔軟な対応が必要」の課題に対応するために、リポジトリを利用します。リポジトリはシステムを開発するすべての資源を集中管理します。また、資源間の関連情報も管理しているので、開発途中での仕様変更や保守局面での変更波及の解析に有効です。

(b) 分散開発作業を支援

SEWB+ のリポジトリでは、チェックイン・チェックアウトや、アクセス権の設定を利用して、複数のグループでの共同作業を支援します。また、エクスポート・インポートによって、ほかのリポジトリに資源を配布することもできます。

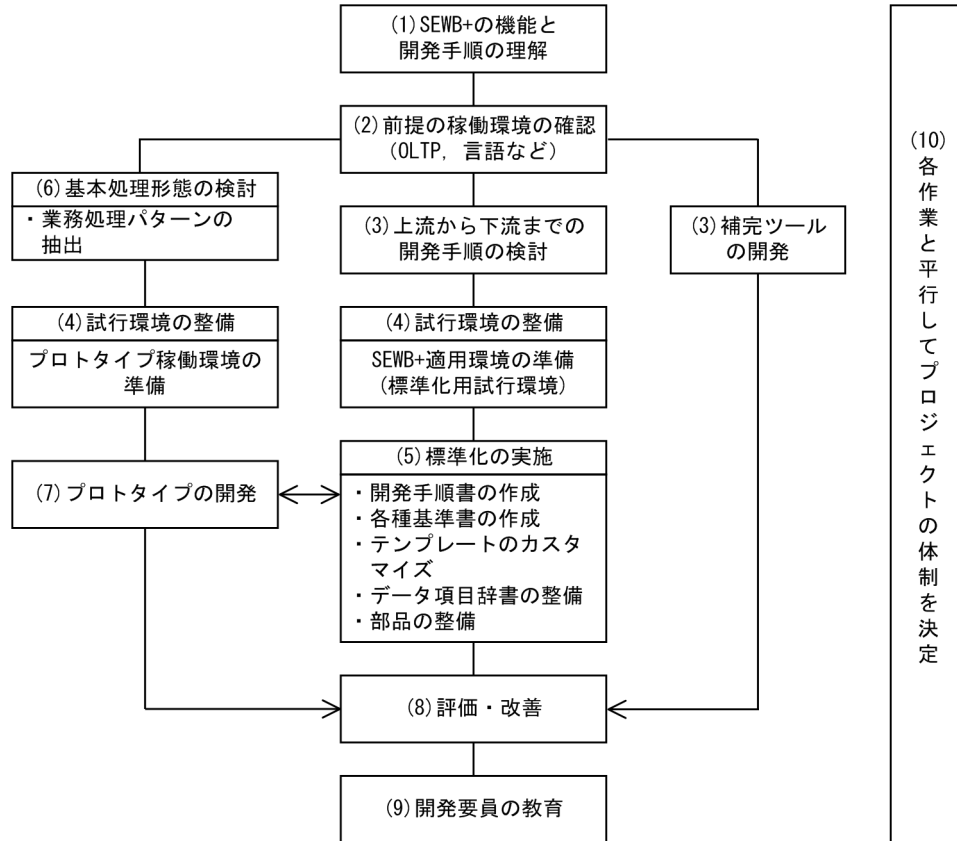
(c) ドキュメント作成支援

保守・納入ドキュメントを作成する工数を削減するため、SEWB+ では、定義内容からのドキュメント印刷をサポートしています。

1.1.2 SEWB+ の導入の流れ

SEWB+ の導入検討から、実際に SEWB+ を利用してのシステム開発までの流れを次の図に示します。ここでは、製品の機能理解から評価・改善、開発要員の教育まで、SEWB+ の導入検討からシステム開発までの作業の流れを示します。なお、フロー中の番号は以降の説明と対応しています。

図 1-2 SEWB+ の導入の流れ



(1) SEWB+ の機能と開発手順の理解

最初に、SEWB+ 全体の機能を把握してください。システムを開発する上で、SEWB+ がサポートしている作業とそのための機能を理解するには、このマニュアルを一読されることをお勧めします。

SEWB+ の開発環境製品は、複数の製品を統合した SEWB+ 基本開発環境セットとして提供しています。SEWB+ 基本開発環境セットは、SEWB+ の開発支援製品である SEWB+ 基本開発環境と、リポジトリの基盤となる Object Server から構成されます。

SEWB+ 基本開発環境の製品の機能とシステム開発の流れについては、「1.4 システム開発の流れ」を参照してください。

(2) 前提の稼働環境の確認

SEWB+ の開発対象 AP を次に示します。詳細は、「1.2 システム構成」を参照してください。

- TP モニタの利用を前提とした C/S システムのサーバ AP
- サーバ上で動作するバッチ処理 AP

(3) 上流から下流までの開発手順の検討

(a) ツールの検討

SEWB+ は、システム開発の下流工程に当たるプログラム設計から作成までを支援するコンポーネント型 CASE ツールです。システムを構築する場合は、下流工程に加え、分析などの上流工程からテストまでを総合的に支援するツール群を整備する必要があります。開発するプロジェクトの特性に応じて、SEWB+ が支援していない部分に対し、必要な支援ツールを検討してください。必要であれば、補完ツールの開発も検討してください。なお、ツールの検討の際には、「1.4 システム開発の流れ」、「2. システムの設計」および「3. DB 設計・データ項目の取り込み」を参照してください。

(b) 辞書の検討

SEWB+ では、辞書をどのように活用するかがポイントになります。辞書にはシステムで使用するデータ項目と、データ項目特有の処理を部品化した業務ルールを登録できます。このため、辞書の利用によってデータ同士の整合性や、生産性および保守性を向上できます。ただし、システムに必要なデータ項目や業務ルールを抽出するには、ある程度の準備期間が必要です。辞書については、「2. システムの設計」および「4. 辞書の作成」を参照してください。

(4) 試行環境の整備

プロトタイプを稼働させたり、標準化をしたりするための試行環境を整備します。なお、開発環境と実行環境は異なる場合があります。試行環境整備の担当には、双方の環境を知っている人を割り当ててください。

(5) 標準化の実施

効率が良いシステム開発には、標準化が必要です。主な標準化事項を次に示します。

開発手順書の作成

SEWB+ を利用したシステム開発の全体の流れと、各工程に必要な作業項目、利用するツール、作成するワークシートなどを決定します。開発人員が多い場合は、手順書を作成して徹底させてください。

各種基準書の作成

代表的な基準を次に示します。

- ネーミングルール（命名ルール）
- 辞書作成・辞書の運用ルール
- 画面・帳票設計基準
- 入出力設計基準
- 仕様変更管理基準
- 開発環境運用基準

テンプレートのカスタマイズ

プロトタイプ開発時に、SEWB+ が提供しているテンプレートにカスタマイズが必要な場合、機能要件に合わせてカスタマイズします。

データ項目辞書の整備

プロトタイプで使用するデータ項目を、データ項目辞書へ登録します。

部品の整備

SEWB+ が提供している部品（DB アクセス用やファイルアクセス用）以外の部品が必要な場合は、部品を開発します。

(6) 基本処理形態の検討

基本的な処理方式を検討してください。処理方式を基に、SEWB+ が提供しているテンプレートをカスタマイズしたり、部品化を検討したりします。主な検討事項を次に示します。

- C/S システムでの、サーバ側のファンクション層とデータ層の切り分け
- 同期制御の方式
- セキュリティの認証方式
- 業務共通処理の抽出

(7) プロトタイプの開発

代表的な業務を抽出して、プロトタイプを開発し、次の評価をしてください。

- 実現性：実際に動くかどうかを確認する
- 性能：性能を見積もる際の基準値の根拠にする
- 生産性
- 開発手順・運用手順の妥当性
- さらに機械化して改善できる作業はないか検討する

ポイント

プロトタイプの開発、およびプロトタイプでの評価はシステム開発上、重要です。必ず実施してください。

(8) 評価・改善

開発手順、標準化の内容および SEWB+ が提供しているテンプレートや部品を総合的に評価して、不具合やさらに効率化をする事項があれば、その手段を決めます。

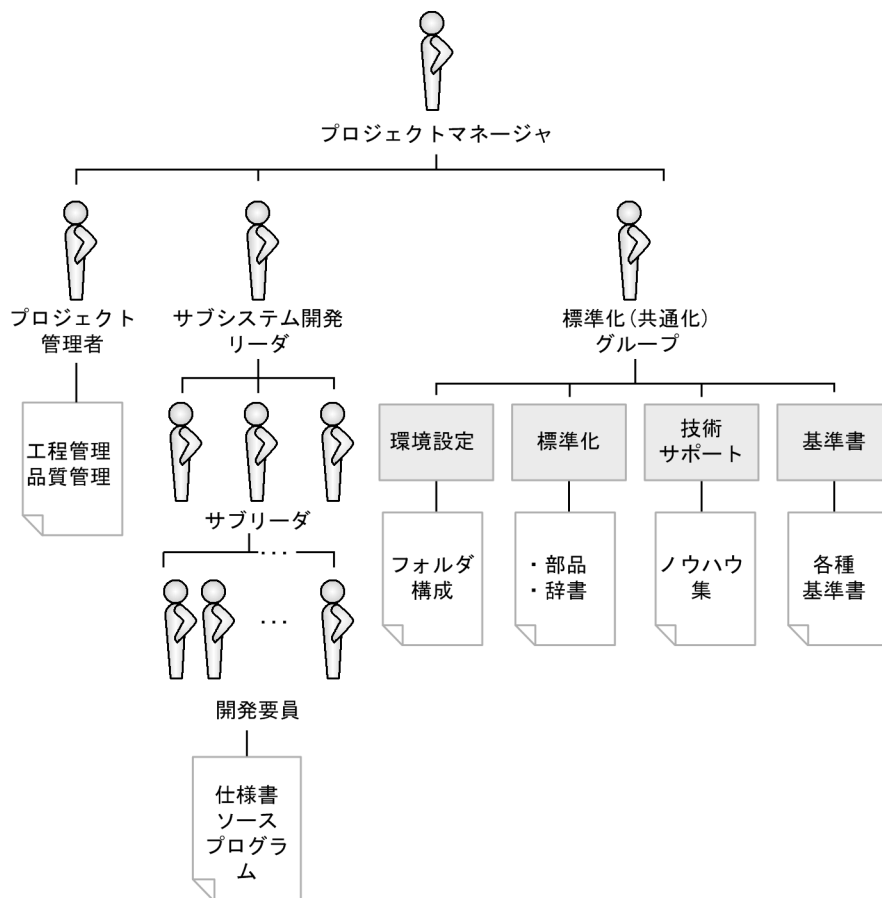
(9) 開発要員の教育

開発担当者向けの教育を実施します。この教育では、一般的なツールの機能説明ではなく、プロジェクト基準や手順を中心にします。プロトタイプ評価で開発したプロトタイプシステムなど、実際のプログラムを利用して、ツールを使いながら教育します。

(10) プロジェクトの体制の決定

導入から開発要員の教育までの作業と並行して、プロジェクトの体制を決めます。大規模なシステム開発の場合、標準化やプロジェクト管理体制を強化することが重要です。プロジェクトの体制の例を次の図に示します。

図 1-3 SEWB+ を適用したプロジェクトの体制の例



1.2 システム構成

SEWB+ は、C/S システム環境の基幹システムを Windows 上で開発するのに適しています。SEWB+ では、次に示すシステムの AP を作成できます。

- C/S システム
- Web システム
- バッチシステム
- ワークフローシステム

なお、AP 作成には SEWB+/CONSTRUCTION を利用します。

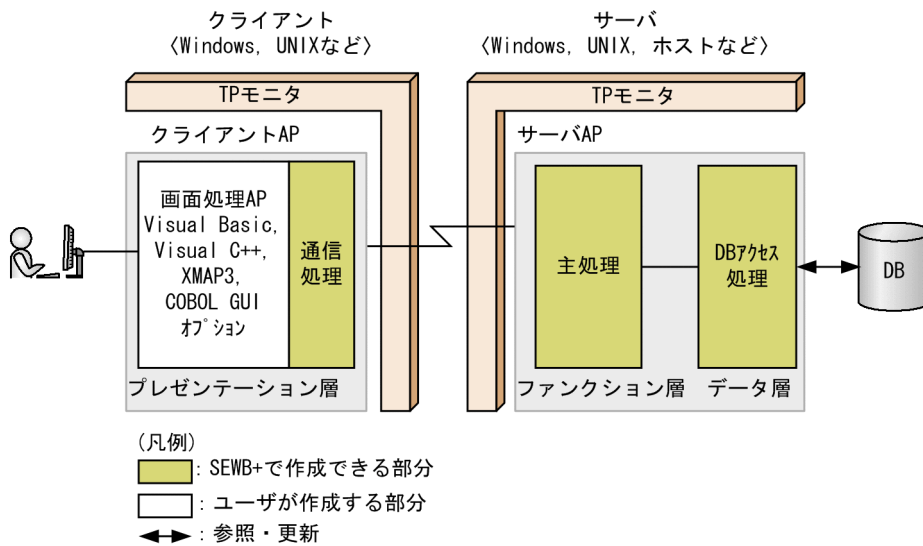
1.2.1 C/S システム

SEWB+ の C/S システムへの適用について説明します。

(1) TP モニタを利用した C/S システム

TP モニタを利用して C/S システム上で動作する AP を作成します。TP モニタを利用した SEWB+ の C/S システムへの適用を次の図に示します。

図 1-4 TP モニタを利用した SEWB+ の C/S システムへの適用



SEWB+ は、次に示す部分の作成に適しています。

- サーバの主処理（ファンクション層）
- サーバのDBアクセス処理（データ層）
- クライアントの通信処理（プレゼンテーション層の一部）

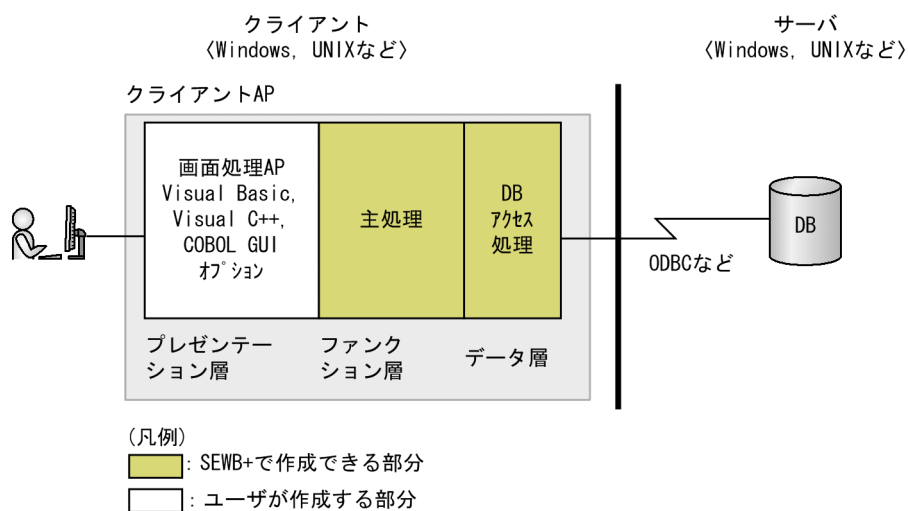
プレゼンテーション層である GUI 画面を設計するときは、Visual Basic、Visual C++、XMAP3、COBOL GUI オプションなどのツールを利用します。

OLTP を使用しないで、独自に通信ロジックを作成する場合も、同様です。

(2) ODBC などを利用した C/S システム

ODBC などを利用した C/S システム上で動作する AP を作成します。作成方法の例として、クライアント側の主処理（ファンクション層）をユーザが作成し、サーバ側の DB とのアクセスは ODBC インタフェースまたは DB のサーバアクセス機能を利用する方法を示します。ODBC などを利用した SEWB+ の C/S システムへの適用を次の図に示します。

図 1-5 ODBC などを利用した SEWB+ の C/S システムへの適用



SEWB+ は次に示す部分の作成に適しています。

- クライアントの主処理（ファンクション層）
- クライアントの DB アクセス処理（データ層）

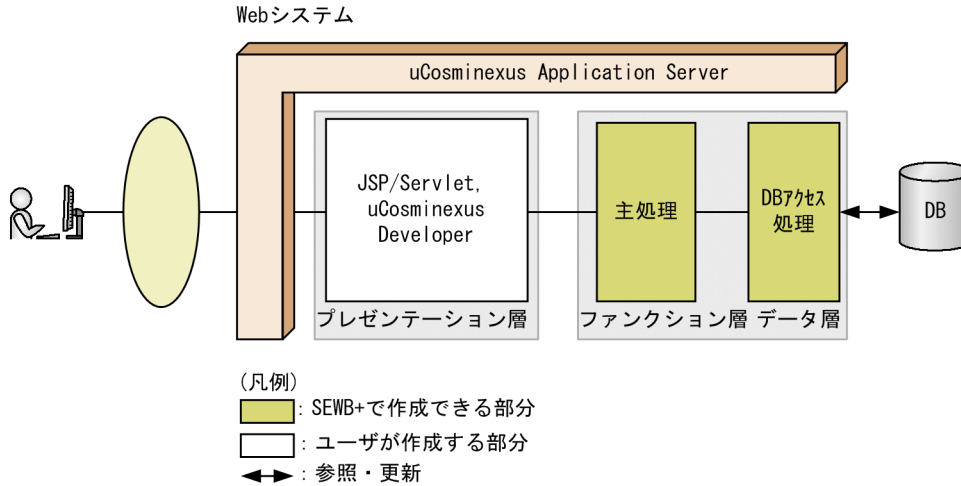
プレゼンテーション層である GUI 画面を設計するときは、Visual Basic、Visual C++、COBOL GUI オプションなどのツールを利用します。

プレゼンテーション層の処理から、同じクライアントにある主処理を呼び出します。

1.2.2 Web システム

Web システム上で動作する AP を作成します。SEWB+ の Web システムへの適用を次の図に示します。

図 1-6 SEWB+ の Web システムへの適用



SEWB+ は、次に示す部分の作成に適しています。

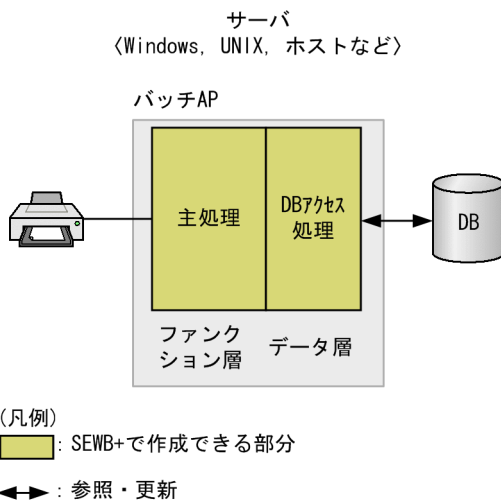
- 主処理 (ファンクション層)
- DB アクセス処理 (データ層)

プレゼンテーション層は、uCosminexus Developer の Page Designer などのツールを利用して作成します。

1.2.3 バッチシステム

SEWB+ では、バッチシステムも対象にしています。SEWB+ を使用して、次に示すような AP を作成します。SEWB+ のバッチシステムへの適用を次の図に示します。

図 1-7 SEWB+ のバッチシステムへの適用



バッチシステム上で動作する AP を作成します。SEWB+ では、バッチシステムの主処理部分 (ファンク

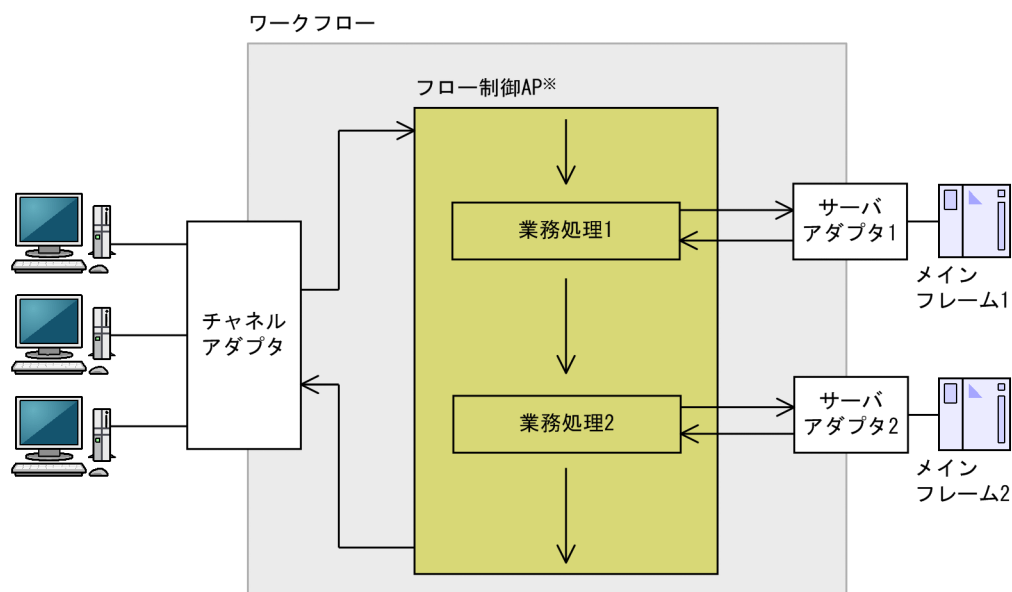
ション層), および DB アクセス処理 (データ層) を作成できます。

帳票の設計は, EUR などのツールを利用してユーザが作成します。

1.2.4 ワークフローシステム

SEWB+ では, 基幹業務をつなぐワークフロー (業務処理の流れ) システムのフロー制御 AP を作成できます。ワークフローシステムとは, 基幹業務の複数のホストやサーバが, アダプタと呼ばれるメッセージ変換プログラムで接続されているシステムです。SEWB+ のワークフローシステムへの適用を次に示します。

図 1-8 SEWB+ のワークフローシステムへの適用



(凡例)

■: SEWB+で作成できる部分

注※ フロー制御APとは, ワークフロー (業務処理の流れ) を制御するプログラムです。次の流れを制御します。

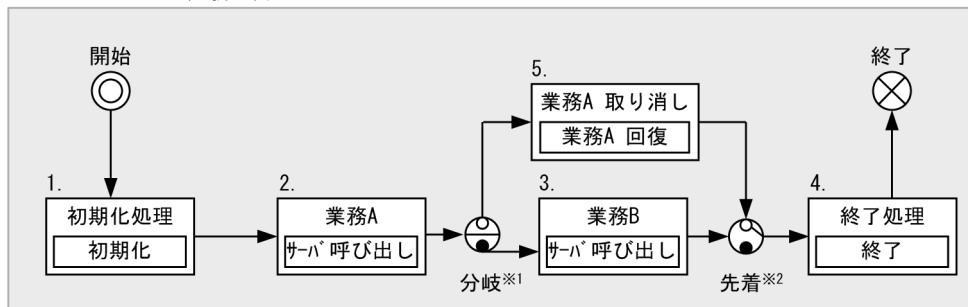
1. 端末に接続しているチャンネルアダプタから要求された必要な業務処理をサーバに依頼する。
2. サーバに依頼した結果に応じて, ほかのサーバを呼び出したり, 結果を元のチャンネルアダプタに返したりする。

ワークフローは, WorkCoordinator Definer を使用して作成します。SEWB+ では, WorkCoordinator Definer が出力するビジネスプロセス定義を取り込んで, フロー制御 AP を作成できます。

WorkCoordinator Definer で作成するビジネスプロセス定義の例を次の図に示します。ビジネスプロセス定義については, マニュアル「WorkCoordinator Definer Version 3 ユーザーズガイド」を参照してください。

図 1-9 WorkCoordinator Definer で作成するビジネスプロセス定義の例

ビジネスプロセス定義の例



注※1 条件に従って分岐します。この場合、異常の有無によって分岐します。
注※2 直前の処理のうち、どちらかが完了した時点で先の処理に進みます。

1. 業務データを初期化する。
2. 業務 A の処理。
3. 業務 B の処理。
4. 業務データの終了処理をする。
5. 業務 A の結果に異常があった場合、業務 A を取り消す。

1.3 SEWB+ の製品

SEWB+ の製品について説明します。

1.3.1 SEWB+ 基本開発環境セット (クライアントサーバ)

SEWB+ のクライアントサーバ型の開発環境をセットにした製品です。

SEWB+ 基本開発環境セットは、SEWB+ の開発支援製品である SEWB+ 基本開発環境と、リポジトリの基盤となる Object Server から構成されます。

SEWB+ 基本開発環境の機能を次の表に示します。

表 1-1 SEWB+ 基本開発環境の機能

SEWB+ 基本開発環境の機能		説明
リポジトリ管理	SEWB+/REPOSITORY	リポジトリを管理する機能です。 開発サーバで稼働します。
	SEWB+/REPOSITORY-BROWSER	開発クライアントからリポジトリの内容を表示する機能 (リポジトリブラウザ) です。
AP 開発支援	SEWB+/CONSTRUCTION	開発クライアントで、AP の構築を支援する機能です。
	SEWB+/RECORD DEFINER	開発クライアントで、AP で使用するレコードの設計を支援する機能です。

1.3.2 SEWB+ 開発クライアント向け製品

SEWB+ 基本開発環境セットで構築したシステムで使用できる、開発クライアント向け製品を次の表に示します。SEWB+ 開発クライアント向け製品については、次に示すマニュアルを参照してください。

- マニュアル「SEWB+/REPORT MANAGER ドキュメント作成支援 使用の手引」
- マニュアル「SEWB+/CODE-DESIGN コード設計支援 使用の手引」
- マニュアル「SEWB+ COBOL 構造図エディタ 使用の手引」
- マニュアル「SEWB+/OLTP DEFINER ユーザーズガイド」

表 1-2 SEWB+ 開発クライアント向け製品

SEWB+ 開発クライアント向け製品		説明
ドキュメント作成支援	SEWB+/REPORT MANAGER	SEWB+ で定義した情報を印刷するための製品です。プログラム処理概要図、インパクトレポートなどを印刷できます。
コード設計支援	SEWB+/CODE-DESIGN	システム設計からプログラム作成までのコード設計作業を支援する製品です。これまで手作業で行ってきたコード表の作成と、コード表に関する COBOL プログラム (コード変換サブルーチン、インタフェーステーブル、コードテーブル) の作成を自動化できます。これによって、作業効率と保守性を向上できます。
COBOL 構造図編集支援	SEWB+ COBOL 構造図エディタ	プログラムの構造をビジュアル化する手法 (PAD) で、COBOL プログラムを表示、編集できます。また、PAD から COBOL プログラムを生成することもできます。
プログラム関連図の作成支援	SEWB+/OLTP DEFINER	プログラム、画面、帳票などのシンボルを配置して、プログラム関連図を作成できます。

1.3.3 SEWB+ ランタイム製品

SEWB+ のランタイム製品を次の表に示します。SEWB+ ランタイム製品については、次に示すマニュアルを参照してください。

- マニュアル「SEWB+/STANDARD-SUBROUTINE 標準サブルーチン 使用の手引」

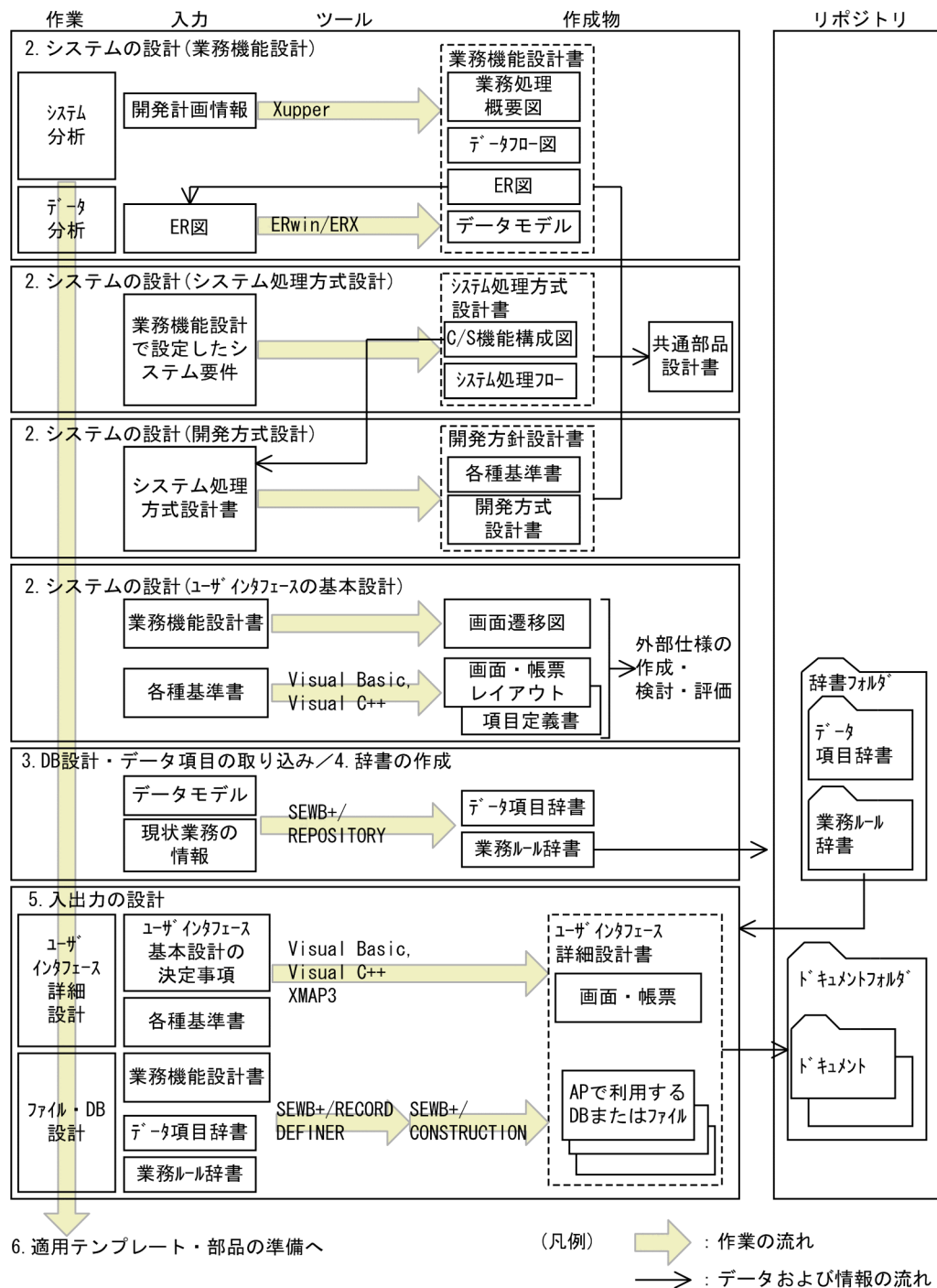
表 1-3 SEWB+ ランタイム製品

SEWB+ ランタイム製品		説明
標準サブルーチン	SEWB+/ STANDARD-SUBROUTINE	標準的なデータ項目に対応する処理（チェック・編集）を実行する部品を COBOL 外部サブルーチンとして提供します。 標準サブルーチンを使用して開発したプログラムを Windows , UNIX で動作させるときに必要です。

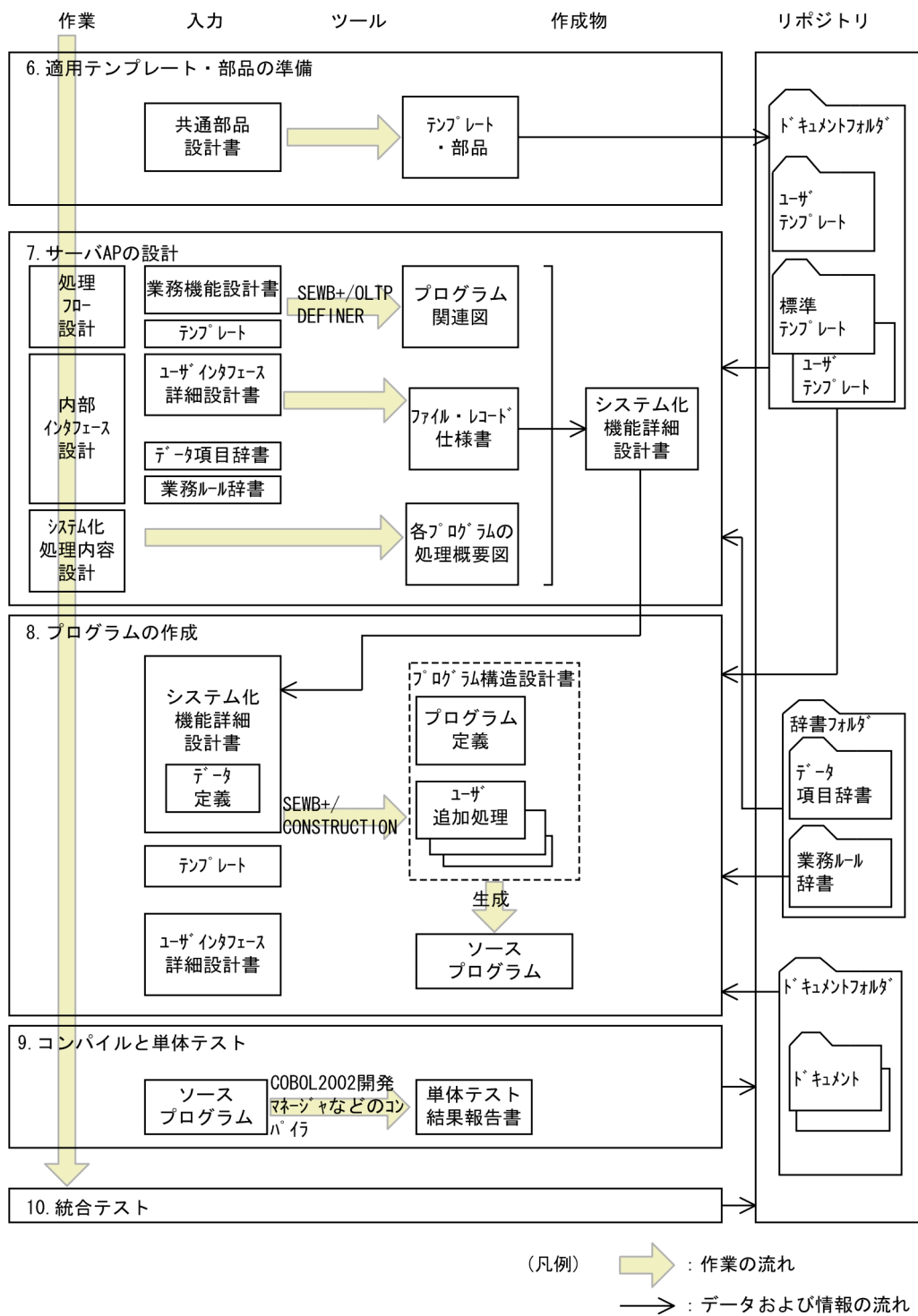
1.4 システム開発の流れ

SEWB+ を利用したシステム開発の流れを次の図に示します。図中の左上に書かれている数字と文字は、詳細を説明している「章番号・章タイトル(補足説明)」を示します。それぞれの手順については、各章を参照してください。

図 1-10 システム開発の流れ (TP モニタを使用)



1. SEWB+ とは



2

システムの設計

システム設計の作業は，システムの外部仕様，内部仕様を決定する工程です。ここでは，TP モニタで SEWB+ を利用したシステム設計について説明します。また，業務の流れや，利用できるツールについても説明します。

2.1 システム設計の概要

2.2 サーバ AP とクライアント AP の処理方式の決定

2.3 開発方式の設計

2.4 データ項目と業務ルールの作成・検討

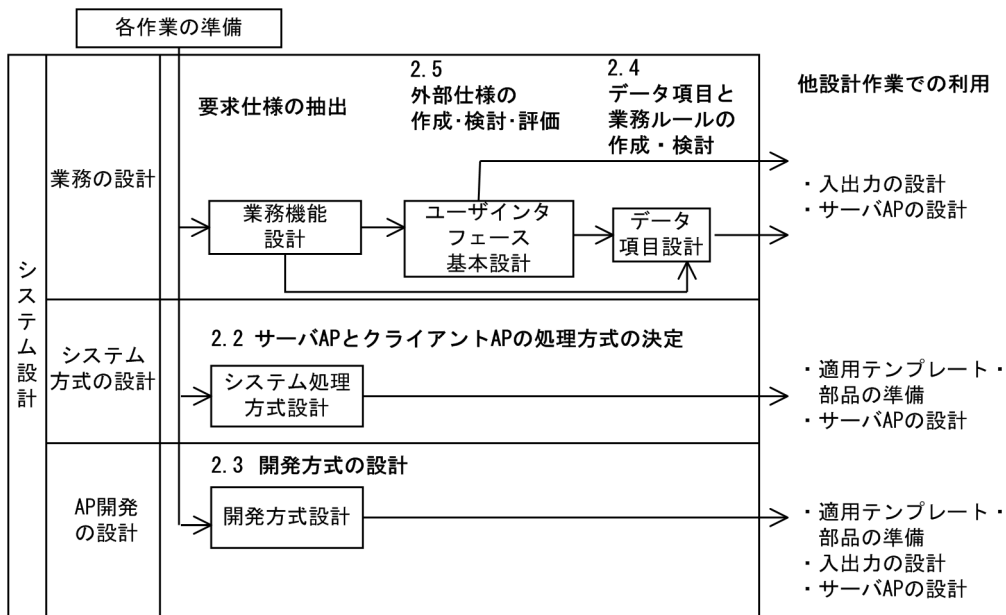
2.5 外部仕様の作成・検討・評価

2.1 システム設計の概要

SEWB+ を利用したシステム設計では、人が処理する作業と機械が処理する作業を考慮して、新業務システムを設計します。それから、手作業での処理と、コンピュータでの処理の切り分けを明確にします。コンピュータが処理する作業については、詳細な要求（画面、帳票、内部処理など）を決定します。システム設計で作成した設計情報は、リポジトリに管理され、AP 作成作業などで利用されます。

システム設計の流れを次の図に示します。この章で説明している手順には、図中に「節番号 節タイトル」を付けています。

図 2-1 システム設計の流れ

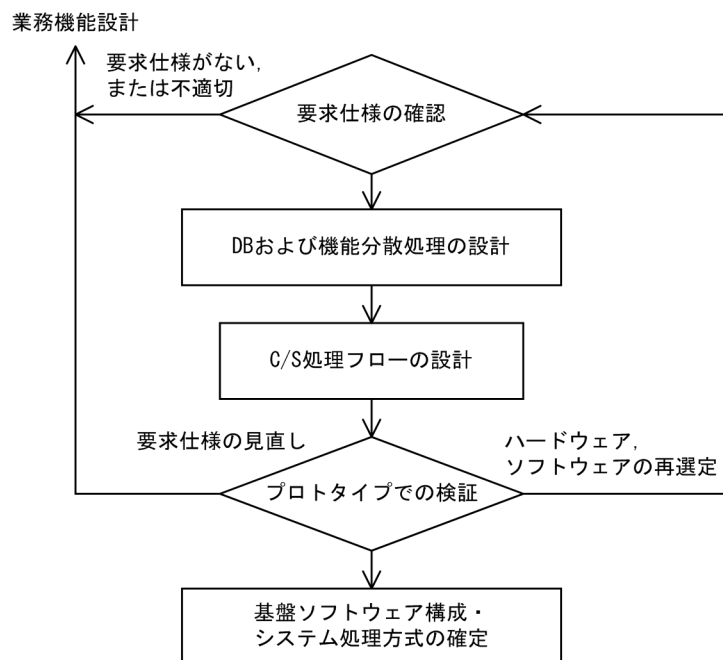


2.2 サーバ AP とクライアント AP の処理方式の決定

業務機能設計で、Xupper、ERwin/ERX などを利用してビジネスフロー図（業務処理概要図）やデータモデルを作成します。システム設計では、この業務機能設計で抽出された要求仕様を基に、OS、GUI および DBMS などのシステムの基盤機能を提供するソフトウェア製品を選定し、それらの製品を前提としてシステム処理方式を設計します。

処理方式を決定する作業の流れを次の図に示します。

図 2-2 処理方式を決定する作業の流れ



2.2.1 要求仕様の確認

すでに業務機能設計などの工程でシステムへの要求が分析されているか確認します。要求仕様が漏れている場合、または内容が不適切な場合は、該当する工程に戻って設計結果を見直してください。

2.2.2 DB および機能分散処理の設計

要求仕様や基盤ソフトウェアの仕様および提供機能を前提にして、DB および各機能を C/S システム上の最適な位置に分散配置します。

(1) 分散化設計の確認

すでに分散化設計書が作成されている場合は、開発対象になる業務に関する DB や、C/S システム上での機能の分散方式を確認します。分散化設計の内容は、C/S 機能構成図にまとめます。

(2) C/S 処理フローの設定

各機能を詳細化し、業務処理ごとに操作およびシステム処理の流れを設定します。C/S 処理フロー設計の内容は、システム処理フロー図にまとめます。システム処理フロー図は、個々のプログラムの処理や入出力データの関連を明確にするシステム化機能詳細設計で利用されます。なお、システム化機能詳細設計に

については「7. サーバAPの設計」を参照してください。

機能を詳細化するときには、次の点に注意します。

- 論理的に「プレゼンテーション層」、「ファンクション層」、「データ層」に区切って設計する。典型的な大規模基幹系システムでの各層は次のようになる。
「プレゼンテーション層」: クライアント側。画面・帳票などの入出力や、簡単なチェックおよび入出力編集などを実行する。
「ファンクション層」: クライアント側またはサーバ側（両方にあってもよい）。主処理の実装部分。複数の層に階層分けできる。
「データ層」: サーバ側。DB だけ、または DB と DB アクセス処理を指す。
- オンライン処理とバッチ処理の切り分け。
- 処理のサイクルおよび処理のタイミング。
- 出力する情報。
- 提供するサービスを受ける処理の切り分け。

2.2.3 設計内容の検証

設計した分散方式や処理の流れで、要求仕様を実現できるかどうかを検証します。

(1) 机上検証

設計したシステム処理方式で不明確な点は、文書などで関連部署や製造元に確認します。

(2) プロトタイプでの検証

設計したシステム処理方式で不明確な点を洗い出し、それらをプロトタイプで確認する必要があるかどうかを判断します。障害回復の処理方式など、運用面からの検証も必要です。

なお、プロトタイプを作成して展開する場合には、次の点に注意してください。

- プロトタイプであること、およびプロトタイプの目的を明確にし、本稼働するシステムと混同させない。
- 本稼働するシステムのハードウェアおよびソフトウェアの購入時期を意識して、プロトタイプでの評価計画を立てる。ハードウェアおよびソフトウェアは、まずプロトタイプ用に 1 セット購入し、評価後に必要台数を手配して各拠点へ展開するとよい。
- 評価するときは、関連製品のバージョンを本稼働させるシステムに合わせるなど、同じ環境を設定する。

プロトタイプの作成については、「2.5 外部仕様の作成・検討・評価」を参照してください。

2.2.4 基盤ソフトウェア構成・システム処理方式の確定

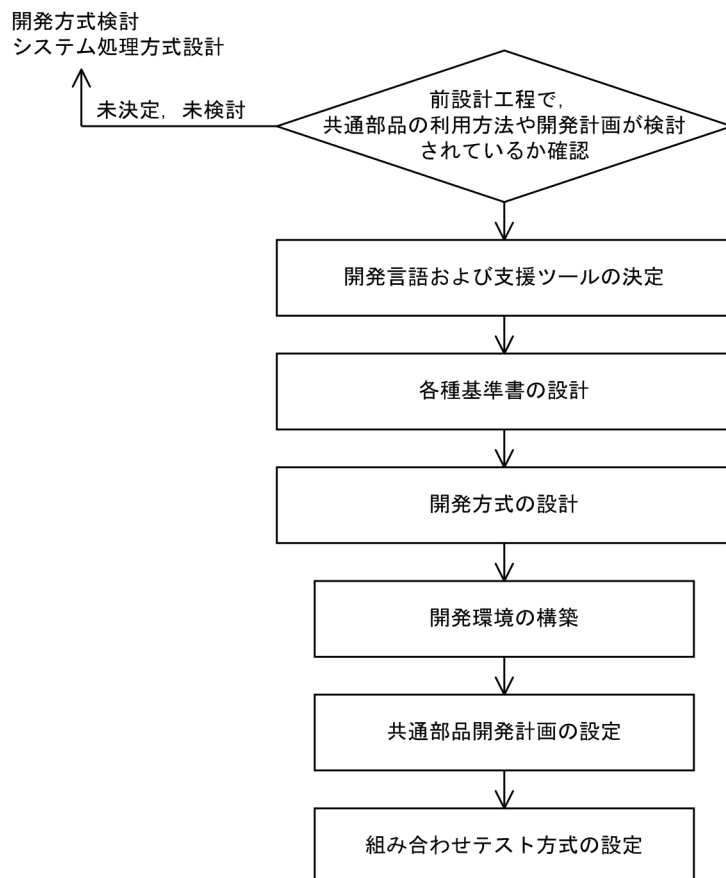
机上検証およびプロトタイプの検証結果を基に、基盤ソフトウェアやシステム処理を確定し、システム処理方式設計書を作成してください。

2.3 開発方式の設計

システム処理方式設計で処理方式が決定したら，その結果を基に C/S システム上の各 AP を開発するための開発方式を設計します。開発方式設計では，開発言語，開発支援ツールおよび開発環境を決定したり，SEWB+ を利用して AP を作成するときに必要なテンプレートや部品の開発計画を決定したりします。

開発方式を設計する作業の流れを次の図に示します。

図 2-3 開発方式を設計する作業の流れ



2.3.1 開発言語および支援ツールの決定

各 AP を開発するための言語および支援ツールを決定します。

2.3.2 各種基準書の設計

AP 開発の手順や基準を設定します。既存の開発標準に準拠して，AP の特性や適用する言語および支援ツールを前提に，プロジェクトの特性に合うようカスタマイズします。また，データモデリングでの名称の標準化もここで見直します。

2.3.3 開発方式の設計

開発方式では、システム処理方式を基に、開発言語を決定したり開発環境を構築したりします。

(1) 開発環境の構築

使用する言語や支援ツール、およびその前提ソフトウェアを含むソフトウェア構成を設定します。また、ソフトウェア導入を前提とした開発用機器をそろえ、メモリ、ディスク容量、デバイスなどのハードウェア構成を見積もります。

(2) 共通部品開発計画の設定

SEWB+ を利用してプログラムを作成する場合に必要なテンプレートおよび部品の開発計画を設定します。テンプレートや部品にするパターンおよび処理は、業務機能設計書やシステム処理方式設計書を基に、各 AP で共通する処理パターンや機能を調査して抽出します。共通部品開発については、「6. 適用テンプレート・部品の準備」を参照してください。

(3) 組み合わせテスト方式設定

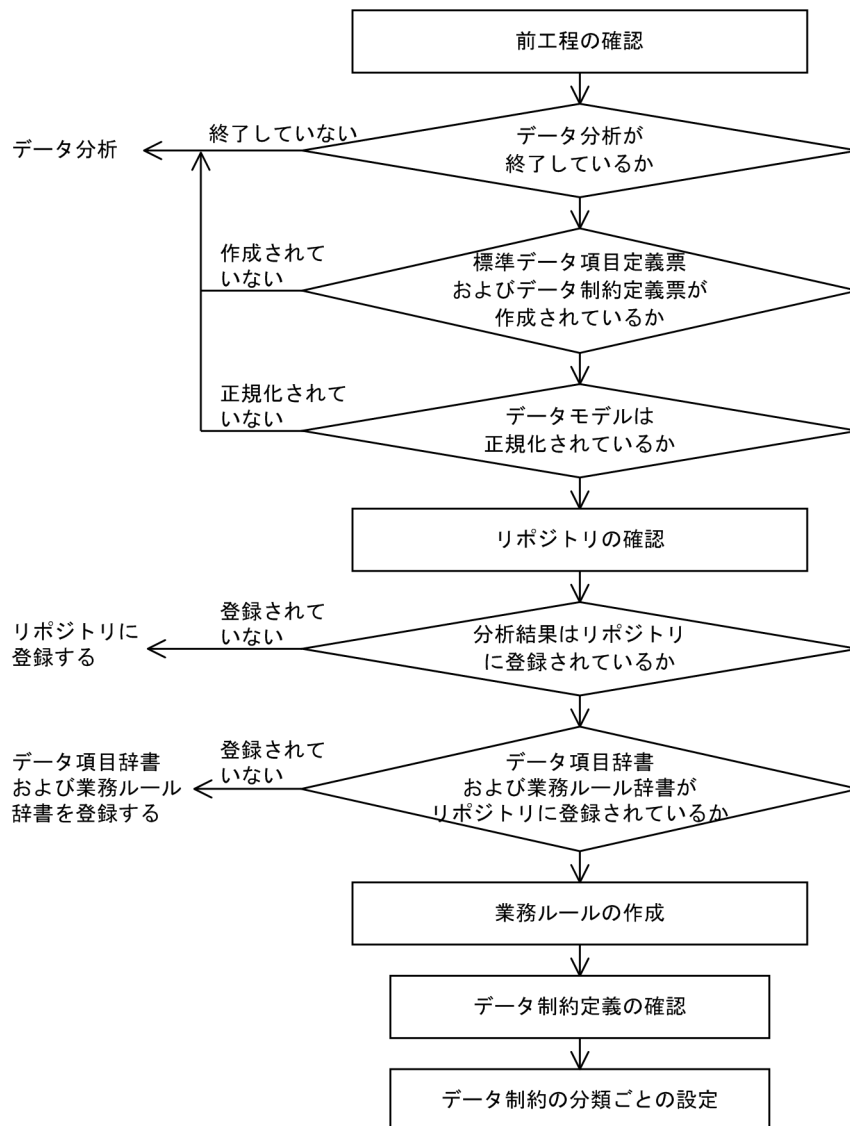
組み合わせテストで実施するテストの範囲、品質目標、終了条件およびテスト実施方法を検討します。また、必要に応じて支援ツールの導入を検討します。

2.4 データ項目と業務ルールの作成・検討

データ分析の工程で Xupper, ERwin/ERX など分析した ER 図やエンティティ定義書を基に, システムで使用するデータ項目を抽出します。これらのデータ項目はデータ項目辞書としてリポジトリに登録します。また, これらのデータ項目に対し, 制約 (ビジネスルール) が指定してある場合, それらは業務ルールとしてリポジトリに登録し, データ項目に適用できるように設定します。

データ項目と業務ルールの作成・検討する作業の流れを次の図に示します。

図 2-4 データ項目と業務ルールの作成・検討する作業の流れ



2.4.1 前工程の確認

前工程として、次に示す作業が終了していることを確認してください。

- データ分析工程で、システムで使用するデータ項目が抽出されていること。
- データモデルが作成されていて、エンティティの詳細情報として、標準データ項目定義票およびデータ制約定義票が作成されていること。
- データモデルが正規化されていること。

2.4.2 リポジトリの確認

リポジトリが次に示す状態であるか確認してください。

- データ分析工程での分析結果がリポジトリに登録されている（ERwin/ERX 使用時は、データ分析結果が CSV 形式ファイルを経由してリポジトリに登録されている）。
- 登録された情報から、必要なデータ項目辞書が作成されている。
- データ項目辞書および業務ルール辞書が登録されている。

2.4.3 業務ルールの作成

業務ルールを作成するときは、次の内容を確認してください。

(1) データ制約定義の確認

データ制約定義票のデータ制約の部分が、業務ルールとして登録されます。データ制約定義票に、データ制約の記述があるかどうかを確認してください。

(2) データ制約の分類ごとの設定

業務ルールは、データ制約の分類ごとに登録内容が異なります。分類は「項目固有か項目関連か」および「チェックか導出か」で決まります。

(a) 分類（データ項目）

項目固有制約

単独のデータ項目に対する業務ルールとして定義します。なお、項目固有制約の業務ルールをドメインデータ項目に定義することで、継承関係を適用できます。詳細は「4.2.1 業務ルールの定義項目」を参照してください。

項目関連制約

複数のデータ項目に関連する業務ルールとして定義します。

(b) 分類（処理のタイプ）

チェック（ルール）

適用するデータ項目の妥当性を判断する業務ルールとして定義します。

導出（ルール）

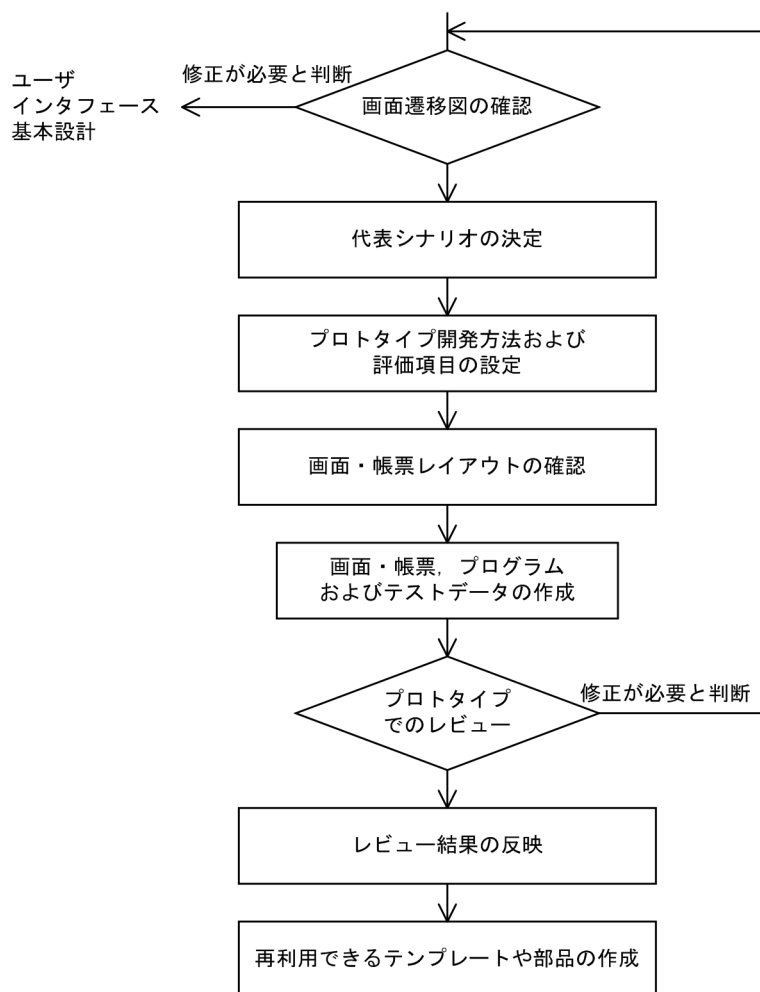
データ項目の値を導き出す業務ルールとして定義します。

2.5 外部仕様の作成・検討・評価

早期に、画面遷移や画面・帳票レイアウトを試験的に操作するプロトタイプを作成し、開発者とユーザとの間の誤解などを解消しておきます。プロトタイプでの評価と検討を繰り返し、早期に外部仕様を固定することで、手戻りのない効率的な開発を目指します。なお、プロトタイプは、ユーザインタフェースの基本設計工程で作成します。ユーザインタフェース基本設計では、外部仕様を決定し、画面・帳票の基準を作成します。詳細なエラー処理や編集処理は、ユーザインタフェース詳細設計で作成します。

外部仕様の作成・検討・評価する作業の流れを次の図に示します。

図 2-5 外部仕様の作成・検討・評価する作業の流れ



2.5.1 画面遷移図の作成

業務機能設計書を基にして画面の流れを画面遷移図にまとめます。

2.5.2 代表シナリオの決定

プロトタイプを作成する範囲と流れ（代表シナリオ）を決定します。

2.5.3 プロトタイプの開発方法および評価項目の設定

プロトタイプを使用したレビューの初期段階では、画面の色や配置などのデザイン面に指摘が偏り、必要なデータや機能の不備が見逃されることがあります。評価の初期段階でこれらの不備を見逃すと、後工程で大きな手戻りが発生してしまいます。評価の初期段階から必要な項目を適切に評価し、DB 構造を早期に確定するために、仕様評価ではデータに関するレビューを優先させます。レビュー対象の優先度は、高い順に「データ」、「機能仕様」、「ユーザインタフェース」となります。

2.5.4 画面・帳票レイアウトの確認

(1) 画面・帳票、プログラムおよびテストデータの作成

プロトタイプの評価では、短期に効率良くレビュー用のプログラムを作成し、レビューを行って問題点を洗い出し、それらを修正することで外部仕様を固めることが目的です。このため、今後、仕様変更が発生しそうな個所や処理が複雑な個所は、ダミーテーブルを利用して、あたかも処理が行われたようにシミュレートします。これらの個所は外部仕様が決まったあと、ユーザインタフェース詳細設計の工程で作り込みます。

(2) プロトタイプでのレビューとレビュー結果の反映

プロトタイプを使用してレビューを行います。指摘や問題点があった場合は、外部仕様を見直します。

2.5.5 再利用できるテンプレートや部品の作成

外部仕様が決まったあと、GUI 基準や処理方式基準（初期表示処理方式、エラー処理方式、部品化など）を見直します。また、再利用できるテンプレートや部品は整備して、以降の開発に利用します。

3

DB 設計・データ項目の取り込み

この章では，上流工程での DOA に基づいたデータ分析・設計，およびツールで作成されたデータモデルの情報を取り込む手順について説明します。

3.1 データ分析とデータ設計

3.2 データ項目の取り込み

3.1 データ分析とデータ設計

SEWB+ で DOA に基づいたデータ項目・業務ルールの定義や管理をするためには、上流工程で DOA に基づいたデータ分析・設計がなされていること、およびそこで分析された結果を適切に SEWB+ に取り込むことが必要です。

SEWB+ は、ERwin/ERX などの標準的なデータ分析ツールや DB 設計ツールと連携し、それらのツールで作成されたデータモデルの情報を取り込みます。

3.1.1 データモデリング

システム全体で使用するデータ項目を、システム開発時に効率良く利用できるようにデータモデルを作成します。

データモデルを作成することをデータモデリングといいます。データモデリングでは、システムで使用するデータ項目を調査したり、それらのデータ項目のドメインを分析したり、名称を標準化したりして、業務システムの静的なデータ項目の構造を、全体像としてまとめていきます。

データモデルはデータ項目などの実態を表すエンティティとエンティティ属性およびエンティティ間の関連であるリレーションシップで構成されます（ER 図）。

大規模システムのデータモデリングでは、これを支援するツールを利用すると効率良く作業できます。データモデリングの進め方については、使用するツールのマニュアルを参照してください。

なお、ERwin/ERX を使用する場合には、作成したデータモデルからエンティティやエンティティ属性の情報を、CSV 形式ファイルを経由して SEWB+/REPOSITORY のデータ項目辞書に登録できます。

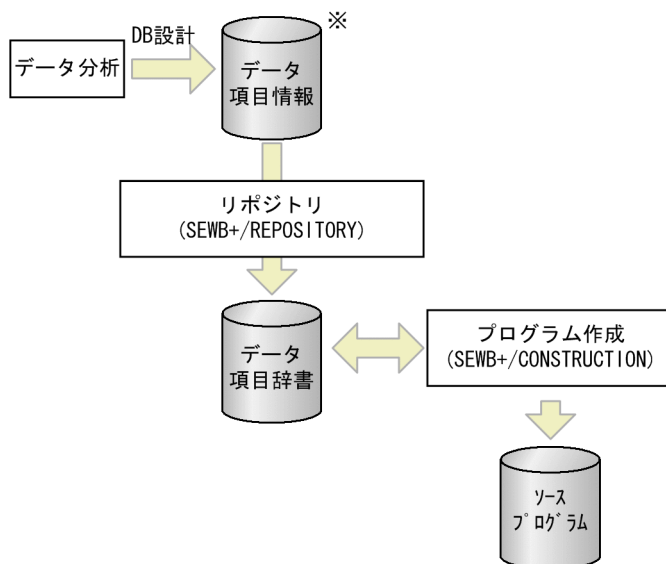
3.2 データ項目の取り込み

SEWB+ では、データ分析や DB 設計で作成したデータ項目をプログラムで利用するために、それらのデータ項目を SEWB+/REPOSITORY に取り込み、データ項目辞書を作成して業務 AP 作成に利用します。データ項目辞書を利用すると、次のような利点があります。

- データ分析時に設計した DB のデータ項目がそのまま使えるので、プログラム作成時にデータ項目を再入力しなくてもよい。
- 記述誤りなどを削減できる。
- データ項目変更時でも、再生成するだけで対処できる。
- データ項目辞書に注目した、変更波及調査ができる。

データ分析からのデータ項目の取り込みを次の図に示します。なお、ERwin/ERX からのデータ項目の取り込み方法については、マニュアル「SEWB+/REPOSITORY 辞書設計ガイド」を参照してください。

図 3-1 データ分析からのデータ項目の取り込み



注※ このデータ項目情報は、次に示すファイルまたは情報を指しています。

- Microsoft ExcelなどのCSV形式ファイル
- ERwin/ERXの出力ファイル
- DBのスキーマ情報

4

辞書の作成

データ分析では、システム全体で使用するデータ項目とその情報を抽出します。SEWB+/REPOSITORYでは、これらを「辞書」に格納して、DOAに基づくシステム開発を実現しています。辞書は、システム開発で扱う情報の基本単位となる「データ項目」と、データ項目特有の処理を部品化した「業務ルール」を管理します。

4.1 データ項目辞書

4.2 業務ルール辞書

4.3 AP 設計と辞書

4.4 辞書使用時の注意事項

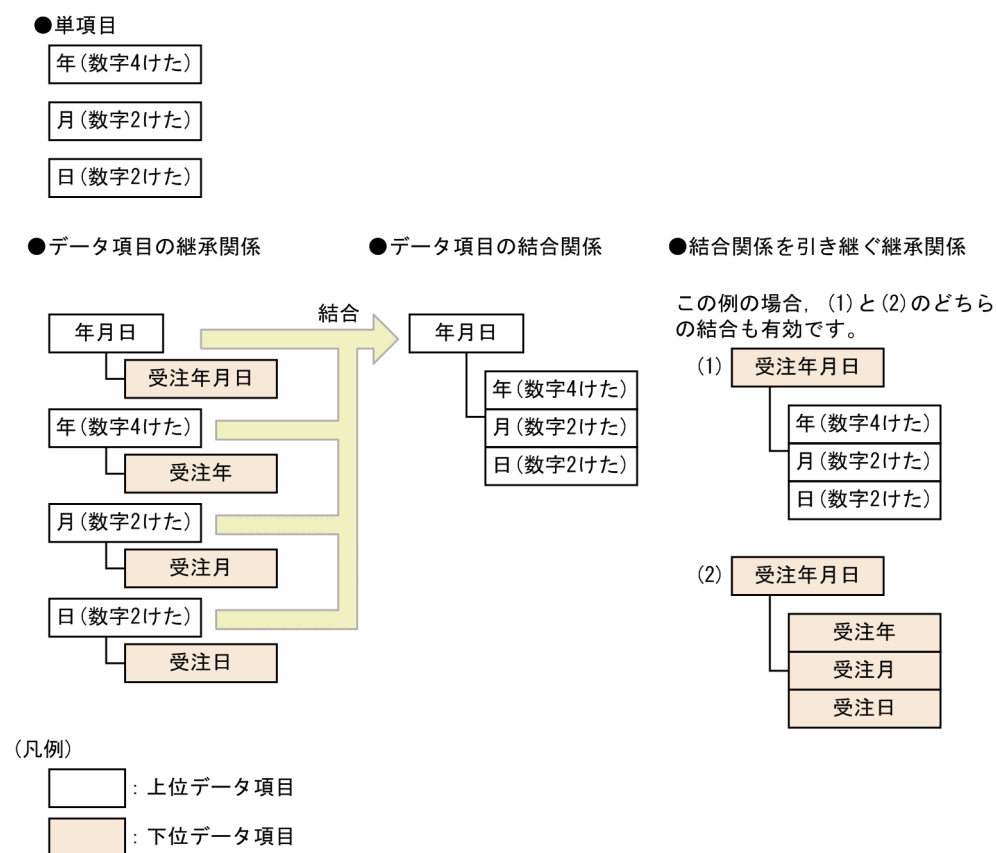
4.1 データ項目辞書

システム全体でデータを共有し、有効活用するためには、データ項目の標準化が必要です。SEWB+ では、データ項目の共有や有効活用を支援するため、DB 設計作業で作成したデータ項目を取り込み、それらを AP で利用できるよう編集・管理する SEWB+/REPOSITORY のデータ項目辞書機能を提供しています。

データ項目辞書では、各データ項目に関する名称（データ項目名、標準名称、フリガナなど）や、属性（データ形式の分類、けた数など）などの言語別詳細情報を定義して、管理します。データ項目辞書で管理するデータ項目には、「単項目」、「継承項目（データ項目の継承関係）」および「結合項目（データ項目の結合関係）」があります。

データ項目辞書の概要を次の図に示します。

図 4-1 データ項目辞書の概要



4.1.1 単項目（基本項目）

データ項目辞書で管理するデータ項目で、ほかのデータ項目と連結していないデータ項目のことです。

単項目は、データ項目辞書の最も基本的なデータ項目の単位です。辞書の設計は、このデータ項目の作成から始めます。データ項目作成時の定義項目の上限や属性などは、プロジェクトごとの利用しやすさを考慮してカスタマイズできます。データ項目の定義のカスタマイズは、SEWB+/REPOSITORY の環境構築ユティリティを使います。環境構築ユティリティについては、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。カスタマイズできる定義項目とカスタマイズ例を次に示します。

- 文字数の上限

- フィールド 1 ~ 20 の名称
例: 「フィールド 1」 「変更履歴」
- 言語区分 4 ~ 7 の追加
例: 「言語 4」 「VB」
- タイプに対応する設定と、レコード生成キーワード
例 (言語区分を「Java」とカスタマイズした場合):
タイプ 「string」
レコード生成キーワード 「string @DATAITEM」

4.1.2 継承項目

データ分析で求められたドメインと、ドメインによってグルーピングされたデータ項目は、継承関係で結び付けてデータ項目辞書に登録します。継承とは、データ項目間で共通する性質を持つデータ項目から、その定義情報を複数のデータ項目に引き継ぐことです。共通する性質を持つデータ項目が継承関係の上位データ項目、引き継ぐデータ項目が継承関係の下位データ項目になります。この継承の仕組みを利用して、ドメインを継承関係の上位データ項目に登録し、グルーピングされたデータ項目を下位データ項目に登録します。このようにドメインの分析結果に基づいてデータ項目に登録すれば、データ項目を効率良く体系的に登録および管理できるとともに、定義情報の標準化を図れます。

(1) ドメインと、ドメインにグルーピングされたデータ項目の登録

ドメインをデータ項目辞書に登録するには、まず最も共通的な値域を持つドメインを、継承関係の最上位のデータ項目として登録します。そのあと、そのデータ項目から定義情報を引き継いで、ドメインにグルーピングされたデータ項目または下位のドメインを登録します。

(2) 継承の利点

ドメインのデータ項目の定義情報は、継承関係の下位データ項目に引き継がれます。したがって、ドメインにグルーピングされたデータ項目には、それぞれのデータ項目固有の定義情報だけを定義し、登録すればよいことになります。

また、継承関係にあるデータ項目を変更する場合、上位のデータ項目を変更すると、継承関係にある下位のデータ項目には、変更が自動的に反映されます。ただし、それぞれのデータ項目固有の定義情報は影響を受けません。このため、変更時の変更波及の管理が容易になります。なお、継承関係にあるデータ項目間でも、ドメインの定義情報を引き継ぐかどうかを設定できます。

4.1.3 結合項目 (レコード項目)

データ分析で求められたデータの構造は、複数のデータ項目同士を結合関係で結び付けて、データ項目辞書に登録します。結合とは、複数のデータ項目に順序を付け、それによってデータ項目を連結させることです。この仕組みを利用して、COBOL の集団項目と基本項目の関係、C 言語の構造体とそれを構成する変数の関係または AP で使用するレコードと、レコードを構成するデータ項目の関係などのデータ構造を、データ項目辞書に登録できます。

データ項目辞書では、結合項目を上位データ項目として継承した場合、上位データ項目の継承項目の結合関係も引き継ぐ定義ができます。

(1) レコード項目 (最上位結合項目) の登録

SEWB+/CONSTRUCTION のデータ定義、EUR でのレポート設計などで利用するレコード項目は、「最上位結合項目」としてデータ項目辞書に登録します。

なお、SEWB+/CONSTRUCTION でのデータ定義では、SEWB+/RECORD DEFINER で作成したレコード定義ファイルを利用することもできます。レコード定義ファイルの使用方法については、マニュアル「SEWB+/RECORD DEFINER 操作ガイド」を参照してください。

(2) AP 利用を考慮したレコード設計

DB やファイル設計、業務設計から求められた C/S システムのインタフェース仕様などの情報を基に、辞書に登録するレコード構造を設計します。このレコード構造に従って、すでに辞書に登録してあるデータ項目を組み合わせる結合項目を構成します。AP でレコードとして扱われる結合項目は、データ項目辞書には「最上位結合項目」として登録します。

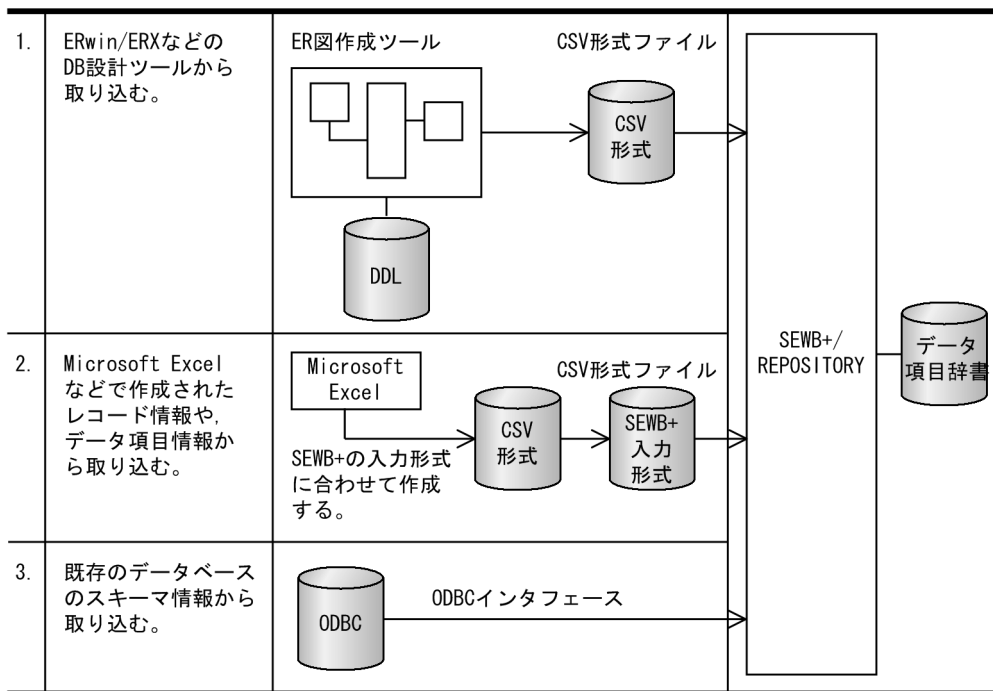
SEWB+/CONSTRUCTION では、実際に AP 中で使用するデータを定義する場合、この最上位結合項目を参照します。例えば、RDB の定義では、RDB の表の構造として最上位結合項目が参照され、データが定義されます。このため、辞書の作成者は、AP 開発に使用するデータの種別を考慮しながら、辞書に用意するレコード構造を決めてください。

4.1.4 データ項目の取り込み方法

DB 設計で作成したデータ項目を AP 作成で使用するためには、まずデータ項目を SEWB+/REPOSITORY のデータ項目辞書に取り込む必要があります。

データ項目辞書にデータ項目を取り込む方法を次の図に示します。データ項目を取り込む手順については、マニュアル「SEWB+/REPOSITORY 辞書設計ガイド」を参照してください。

図 4-2 データ項目を SEWB+ に取り込む方法



1. 業務分析やデータ分析をして、ERwin/ERXなどでDB論理設計をする場合に適用する方法。
2. ER図にないデータ項目を管理する場合などに適用する方法。
自由度の高いデータ項目管理方式が要求されている場合に適します。
3. すでに構築されているDBから情報を取り込むときに適用する方法。

注 ER図が変更された場合は、ER図情報の取り込みからやり直してください。

4.2 業務ルール辞書

業務ルールとは、データ項目に着目して、データ項目特有の処理を部品化したものです。データ項目辞書を整備したあと、業務ルールを格納する業務ルール辞書を整備します。

業務ルールには、通常の業務ルールと、同じデータ項目を使った処理を記述するための同一項目用業務ルールがあります。通常の業務ルールには、単独のデータ項目に対する処理を定義する業務ルール（項目固有ルール）と複数のデータ項目に関連した処理を定義する業務ルール（項目関連ルール）を定義できます。また、項目固有ルールは、複数のデータ項目間で共有することもできます。

4.2.1 業務ルールの定義項目

業務ルールを定義する場合、業務ルールの処理内容（ルールスクリプト）、業務ルールの名称、および業務ルールとデータ項目の関連を定義します。業務ルールとデータ項目との関連は、データ分析工程で抽出したデータ制約の種別によって定義内容が異なります。データ制約は「項目固有制約」と「項目関連制約」に分類されます。また、それぞれの処理内容によって「チェックルール」と「導出ルール」に分類されます。

項目固有制約

単独のデータ項目に対する業務ルールとして定義します。定義する場合は、SEWB+/REPOSITORYの[業務ルール新規作成]ダイアログの[データ項目]タブで、「適用データ項目」の「専用にする」をチェックし、「組み合わせるデータ項目」に、この業務ルールを適用するデータ項目を指定します。このとき指定できるデータ項目は一つだけです。

ドメイン制約（項目共通制約）

ドメインとしてグルーピングされ、辞書に登録されているデータ項目との間に継承関係を定義する場合、ドメインに対して業務ルールを定義します。この場合、定義方法は項目固有制約と同じです。

項目関連制約

複数のデータ項目に対する業務ルールとして定義します。定義する場合は、SEWB+/REPOSITORYの[業務ルール新規作成]ダイアログの[データ項目]で、「適用データ項目」の「専用にする」をチェックし、「組み合わせるデータ項目」に、この業務ルールを適用するデータ項目を指定します。このとき指定できるデータ項目は一つ以上です。

(1) 分類（処理のタイプ）

チェック（ルール）

適用するデータ項目の妥当性を判断する業務ルールとして定義します。定義する場合は、SEWB+/REPOSITORYの[業務ルール新規作成]ダイアログの[データ項目]で、「適用データ項目」の「組み合わせるデータ項目」の「適用条件」に「入力」を指定します。

導出（ルール）

データ項目の値を導き出す業務ルールとして定義します。定義する場合は、SEWB+/REPOSITORYの[業務ルール新規作成]ダイアログの[データ項目]で、「適用データ項目」の「組み合わせるデータ項目」の「適用条件」に次のように指定します。

- 業務ルールによって導き出される値を格納するデータ項目：出力
- 業務ルールを適用するそのほかのデータ項目：指定なし

(2) 属性

「業務ルール名」、「標準名称」、「フリガナ」、「コメント」および「フィールド1～3」のことです。「業務

4. 辞書の作成

ルール名」、「標準名称」および「フリガナ」は業務ルールを識別しやすいように定義します。「コメント」は処理概要やソースプログラムへの展開条件、注意事項などを記述します。「フィールド 1 ~ 3」は、プロジェクト独自の情報を定義するなど自由に使用できます。

(3) データ項目との関連

業務ルールとデータ項目とを関連づけるため適用データを専用にするか共用にするかや、適用条件、ルールスクリプトで使用するキーワードなどを定義します。

(a) 適用データの専用と共用

業務ルールが特定のデータ項目だけに対応するのか、複数のデータ項目に対応するのかを選択します。複数のデータ項目に対応する必要がある場合、ドメインに業務ルールを対応させて適用を継承させる、継承関係を利用することをお勧めします。

(b) キーワード

業務ルールに使用するデータ項目の仮の名称です。

(c) 適用条件

データ項目がどのような条件（処理の入力または出力）にあるときに、その業務ルールを AP に適用させるのかという情報を指定します。適用条件は、プログラム作成に必要な業務ルールを SEWB+/CONSTRUCTION で抽出する場合にキーとなる情報です。したがって、SEWB+/CONSTRUCTION での利用も考慮した指定が必要になります。

(4) ルールスクリプト

ソースプログラムに展開する業務ルールの処理を、AP 開発に使用するプログラミング言語別にコーディングします。

4.2.2 業務ルールの登録

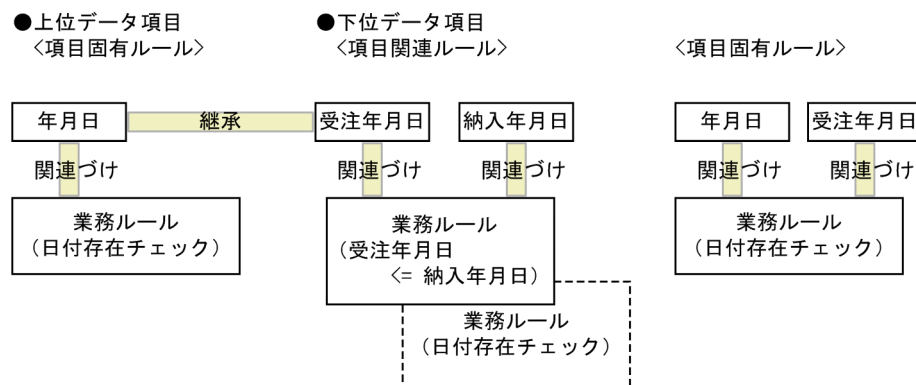
業務ルールを登録する方法について説明します。

(1) 継承関係を利用した登録

業務ルールは、継承できます。業務ルールに継承関係の上位データ項目を関連づけると、その業務ルールは下位データ項目にも適用されます。

継承関係を利用した登録を次の図に示します。

図 4-3 継承関係を利用した登録



業務ルールを整備するときは、継承の関係を考慮してください。ただし、同一項目用業務ルールの設計時には、継承の関係を考慮する必要はありません。

(2) システムで共用する業務ルールの登録

実際に業務ルール辞書を整備するには、まず、ドメイン、つまり継承関係の上位データ項目の業務ルールから登録を始めます。ドメインと業務ルールの関連は、データ項目の体系と同じようにドメインを継承するすべてのデータ項目に引き継がれ、業務ルールを共用できます。したがって、ドメインの業務ルールは、システムで共用することを目的とした、汎用性の高い業務ルールといえます。

ドメインの業務ルールは、次の手順で登録します。手順については、マニュアル「SEWB+/REPOSITORY 辞書設計ガイド」を参照してください。

1. データの値の妥当性をチェックする業務ルール
2. データの値を導き出す業務ルール

(3) 特定業務で扱う業務ルールの登録

ドメインの業務ルールを業務ルール辞書に登録したあと、継承関係の下位データ項目の業務ルールを登録します。特定のデータ項目に関する業務の規則、制度などを業務ルールとして登録します。特定業務を扱う業務ルールも、システムで共用する業務ルールと同様の手順で登録してください。

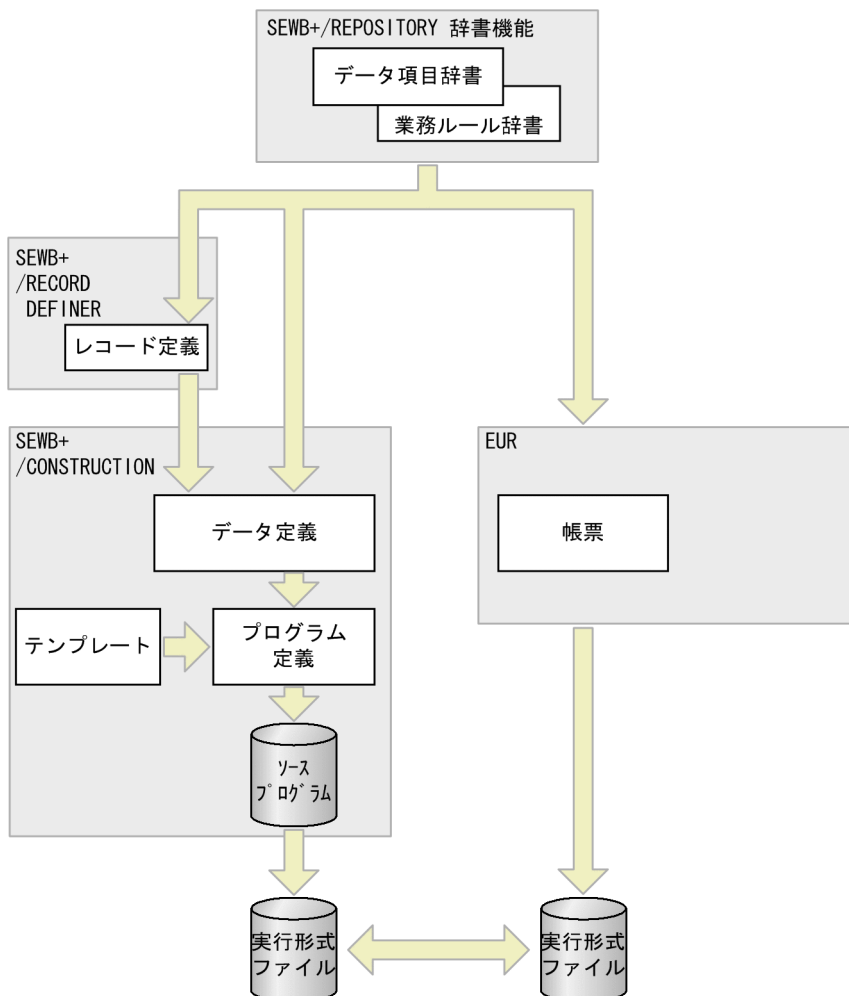
4.3 AP 設計と辞書

SEWB+/REPOSITORY の辞書機能を利用した AP 設計では、上流のデータ分析工程の情報を辞書に取り込んで一括管理するため、次の利点があります。

- データ分析時に設計した DB のデータ項目がそのまま使えるので、プログラム作成時にデータ項目を再入力しなくてもよい。
- 記述誤りなどを削減できる。
- データ項目変更時でも、再生成するだけで対処できる。
- データ項目辞書に注目した、変更波及調査ができる。

辞書を基にした AP 設計を次の図に示します。

図 4-4 辞書を基にした AP 設計



4.4 辞書使用時の注意事項

SEWB+ の辞書を使用するときの注意事項について説明します。

(1) COBOL2002 使用時の注意

COBOL2002 を使用する場合、次のことに注意してください。

- REDEFINES 句は使用できない。
- プログラム中で使用する FILLER も、あらかじめ辞書に登録しておく。

注

SEWB+/RECORD DEFINER でレコード定義ファイルを作成する場合には、REDEFINES 句、FILLER を定義できます。

(2) 辞書変更での注意事項

辞書を変更するときの注意事項について説明します。

(a) 辞書の状態保存

辞書は自動的に最新の状態になるので、履歴を採る必要はありません。データ定義など、辞書参照で作成するドキュメントをある時点の状態のまま保存しておきたい場合は、次のどちらかを実施してください。

- ドキュメントを参照専用でチェックアウトして、ローカルで保存する。
- SEWB+/CONSTRUCTION の環境設定で「データ定義およびレコード定義で辞書の自動再入力をしない」を設定して操作する。

(b) プログラム生成への影響

データ定義で使用している辞書情報に変更があった場合、それらを取り込んだプログラム定義でプログラム生成すると、最新のデータ項目情報を取得するため時間が掛かることがあります。ソースプログラムの生成時間を短縮するために、次の操作を定期的に行うことをお勧めします。

データ定義の更新方法

1. ローカルマシンに更新専用でチェックアウトする。
2. リポジトリにチェックインする。

(3) データ定義情報生成時の注意

SEWB+/CONSTRUCTION の環境設定で「辞書を参照しないでデータ定義情報を生成する」を設定すると、辞書を参照しないでデータ定義情報を生成します。データ定義ファイルがチェックインされている場合またはプログラム定義で「辞書の利用」を指定している場合でも、辞書を参照しないで、データ定義情報が生成されるので注意が必要です。

5

入出力の設計

SEWB+ は、Visual Basic などの GUI 構築ツールと連携して、ユーザインタフェースを設計できます。また、データ項目辞書の最上位結合項目を取り込んで、AP で使用するファイルまたは DB を定義します。

5.1 ユーザインタフェースの設計

5.2 ファイル・DB の設計

5.1 ユーザインタフェースの設計

SEWB+ は、ユーザインタフェースを設計する GUI 構築ツールや帳票設計ツールと連携できます。ユーザインタフェース部分は、エンドユーザがデータ項目を入力したり、コンピュータが処理結果を出力したりする入出力部分です。

GUI 構築ツールには XMAP3 および Visual Basic を、帳票設計ツールには EUR を使用できます。これらのツールを使用して画面や帳票のレイアウトを設計し、作成したプロトタイプでインタフェースを検証できます。このように、プロトタイプで画面や帳票のレイアウトを確認することで仕様を早期に確定させ、設計の手戻り作業を防止するようにします。

5.1.1 GUI 構築ツールでのクライアント AP の設計

GUI 構築ツールとして、XMAP3 および Visual Basic をクライアント AP とした、C/S システムを構築できます。

(1) XMAP3 を使用した場合

XMAP3 は、レイアウト定義ウィンドウにテキストやボタンなどのオブジェクトを置くことで、表現力豊かな画面・帳票を簡単に作成できるツールです。画面・帳票を作成するには、XMAP3 のドロー機能を使用します。作成した画面・帳票の定義情報は、マップ定義ファイルとして保存されます。さらに、マップ定義ファイルをクライアント AP 作成時に SEWB+/CONSTRUCTION で使用すると、画面との整合性を取れます。

XMAP3 での画面・帳票の作成方法については、次に示すマニュアルを参照してください。

XMAP3 Version 3, XMAP3 Version 4

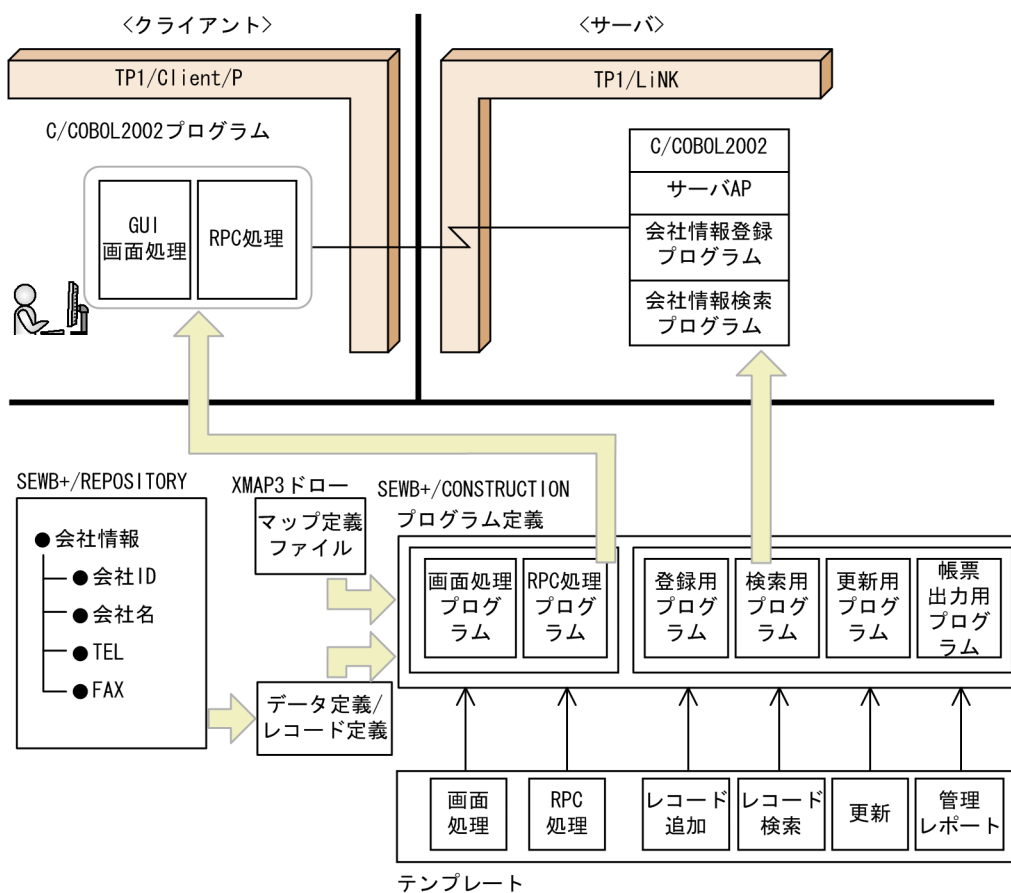
「XMAP3 プログラミングガイド 画面編」または「XMAP3 プログラミングガイド 帳票編」

XMAP3 Version 5

「XMAP3 開発ガイド」または「XMAP3 プログラミングガイド」

XMAP3 を使用したクライアント AP の設計手順を次の図に示します。

図 5-1 XMAP3 を使用したクライアント AP の設計手順



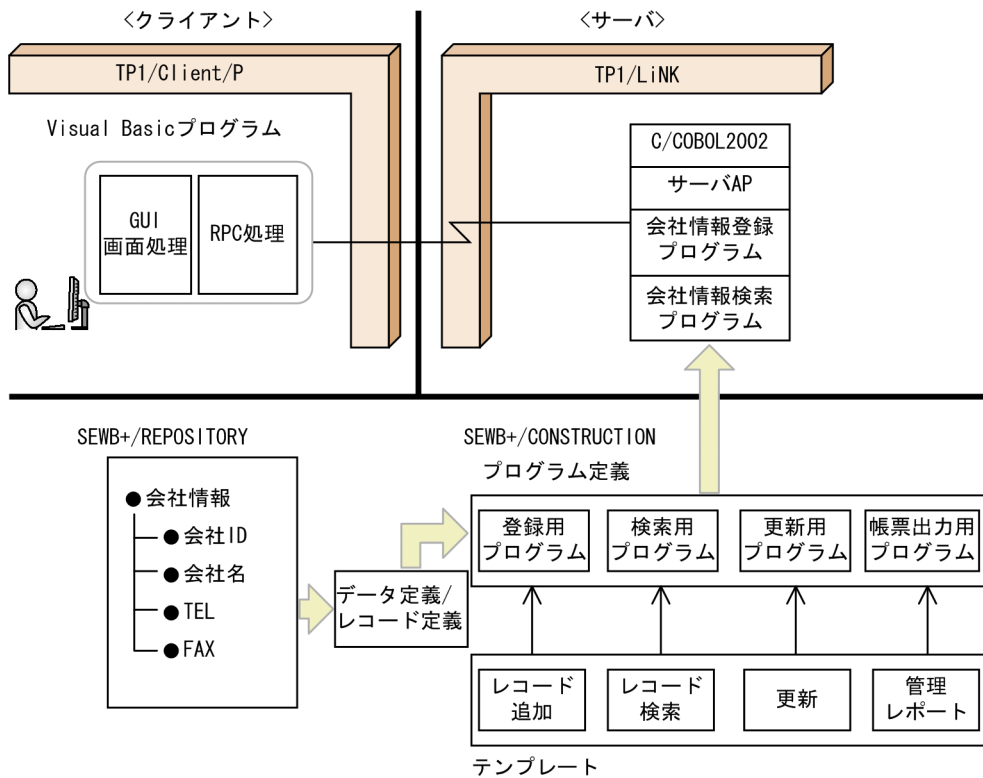
(2) Visual Basic を使用した場合

Visual Basic は、フォームの上にテキストやコマンドボタンなどの部品を置くことで、インタフェースを作成します。さらに、部品に対してプロパティを設定したりコードを記述したりして、GUI を設計します。

ただし、RPC に関する処理や、クライアント AP とサーバ AP のレコード形式を合わせる作業は、データ項目辞書を基にユーザが行う必要があります。

Visual Basic を使用したクライアント AP の設計手順を次の図に示します。

図 5-2 Visual Basic を使用したクライアント AP の設計手順



5.1.2 帳票レイアウト設計

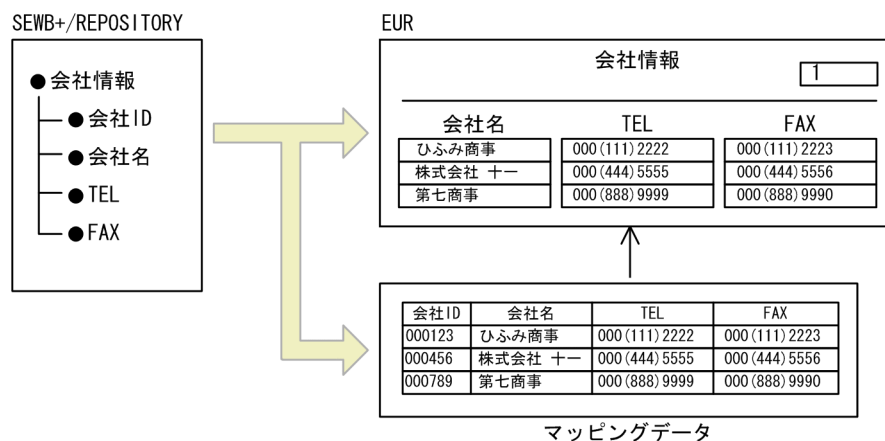
帳票設計ツールとして EUR を使用して、帳票レイアウトおよび各フィールドの仕様を設計します。

EUR は、SEWB+/REPOSITORY のデータ項目辞書と連携して、帳票を設計できます。データ項目辞書を使用すると、サーバ AP とのインターフェースの整合性を取れます。

また、設計した帳票レイアウトは印刷機能で印刷できるので、エンドユーザとの仕様確認などに利用できます。帳票設計の手順については、「7.2 帳票出力する AP の設計」を参照してください。

EUR を使用した帳票レイアウトの設計を次の図に示します。

図 5-3 EUR を使用した帳票レイアウトの設計



5.2 ファイル・DB の設計

システムで利用するデータ項目は、データ分析の工程ですでに抽出されています。データ分析で抽出されたデータ項目は、データ項目辞書としてリポジトリに登録されます。

ここでは、データ項目辞書を利用してファイルまたはDBを設計し、APで利用する入出力項目を決定します。

5.2.1 レコード定義

APで使用するファイルやDBを定義するには、SEWB+/CONSTRUCTIONのデータ定義機能を利用します。リポジトリでは、すでに辞書作成の段階で、SEWB+/CONSTRUCTIONのデータ定義を考慮してレコード構造が設計され、そのレコード構造に従ったデータ項目の組み合わせが結合項目として登録してあります。また、結合項目の中で、レコードとして定義されるものは「最上位結合項目」として登録されています。SEWB+/CONSTRUCTIONでのデータ定義では、最上位結合項目を取り込みます。

なお、SEWB+/CONSTRUCTIONでのデータ定義では、SEWB+/RECORD DEFINERで作成したレコード定義ファイルを使用することもできます。詳細は、マニュアル「SEWB+/RECORD DEFINER 操作ガイド」を参照してください。

(1) SEWB+/CONSTRUCTIONでのファイルおよびDB定義

SEWB+/CONSTRUCTIONのデータ定義では、データ定義種別ごとに必要な情報を定義します。この工程で定義するデータ定義種別は、「ファイル」、「DAM」、「TAM」および「RDB」です。

ファイル

ファイルの入出力に必要な「ファイル名」、「ファイル編成」および「レコード構造」を定義します。「レコード構造」は、データ項目辞書から最上位結合項目、またはSEWB+/RECORD DEFINERで作成したレコード定義ファイルを選んで定義します。

DAM

DAMファイルの入出力に必要な「DAMファイル名」、「入出力ブロックサイズ」および「レコード構造」を定義します。「レコード構造」は、データ項目辞書から最上位結合項目、またはSEWB+/RECORD DEFINERで作成したレコード定義ファイルを選んで定義します。

TAM

TAMテーブルの入出力に必要な「TAMテーブル名」および「レコード構造」を定義します。「レコード構造」は、データ項目辞書から最上位結合項目、またはSEWB+/RECORD DEFINERで作成したレコード定義ファイルを選んで定義します。

RDB

RDBの入出力に必要な「表名称」および「レコード構造」を定義します。「レコード構造」は、データ項目辞書から最上位結合項目、またはSEWB+/RECORD DEFINERで作成したレコード定義ファイルを選んで定義します。

5.2.2 ファイル・DB 設計時の注意事項

SEWB+ でのファイルまたは DB 設計時の注意事項について説明します。

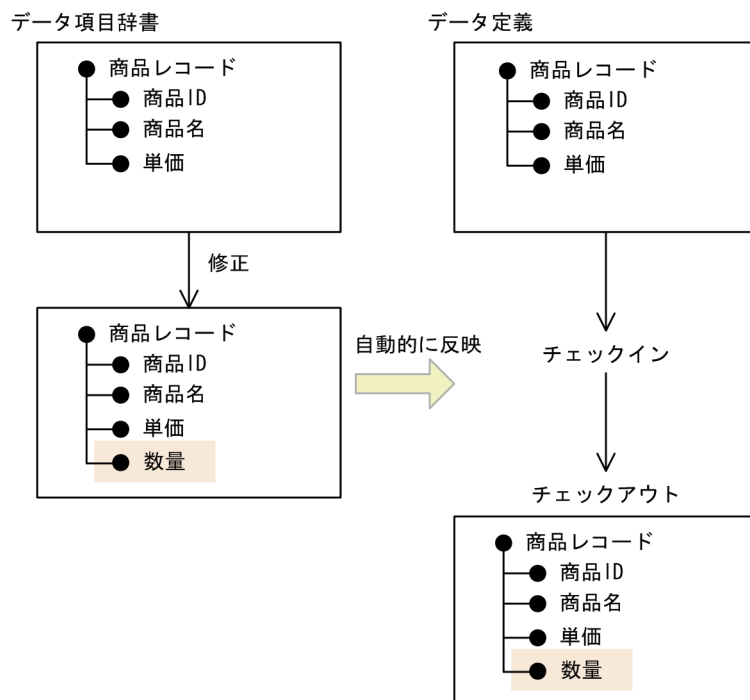
(1) データ定義の注意事項

SEWB+ の辞書は常に最新状態になるよう、自動的に更新されます。このため、データ定義の内容も常に最新状態を反映します。したがって、辞書を使用したデータ定義は常に最新状態なので履歴を採る必要はありません。ただし、データ定義の状態を保存しておきたい場合は、次のどちらかを実施してください。

- データ定義のドキュメントを参照専用でチェックアウトして、ローカルで保存する。
- SEWB+/CONSTRUCTION の環境設定で「データ定義およびレコード定義で辞書の自動再入力をしない」を設定して操作する。

データ項目辞書の修正と SEWB+ の自動更新の関係を次の図に示します。

図 5-4 データ項目辞書の修正と SEWB+ の自動更新の関係



(2) COPY 文と @@expand 文の機能比較

レコードは、ソースプログラム中に書く COPY 文またはテンプレート記述言語の @@expand 文で生成ソースに展開できます。COPY 文と @@expand 文で展開したときの違いを次の表に示します。

表 5-1 COPY 文と @@expand 文の機能比較

相違点	展開方法	
	ソースプログラム中に書く COPY 文	@@expand 文
取り込みのタイミング	COBOL ソースのコンパイル時	COBOL ソースの生成時
取得できる関連	<ul style="list-style-type: none"> • データ定義とデータ定義から生成したレコードソースの関連 • データ定義とデータ項目 	プログラム定義、ソースプログラム、データ定義、レコード定義およびデータ項目間の関連を一度に取得できる。

5. 入出力の設計

相違点	展開方法	
	ソースプログラム中を書く COPY 文	@@expand 文
データ定義変更時	該当するデータ定義の生成だけ必要	データ定義を参照する，すべてのプログラム定義でのソース再生成が必要
COBOL 開発マネージャでの修正方法	データ定義を開いて，生成してからビルドする（データ定義変更時）。	ビルドによって，データ定義の変更内容（データ項目の変更を含む）は自動的に反映される。
ターゲットへの反映手順	生成したレコードソースだけターゲットシステムへ転送し，ターゲットシステムで取り込んでいる COBOL ソースを再コンパイルする。	該当する COBOL ソースをすべて転送し，COBOL ソースを再コンパイルする。

6

適用テンプレート・部品の準備

複数の AP の共通部品として調査・抽出されたパターンや処理を，AP 開発で必要なテンプレートおよび部品として作成しておきます。ここでは，テンプレートおよび部品の作成について説明します。

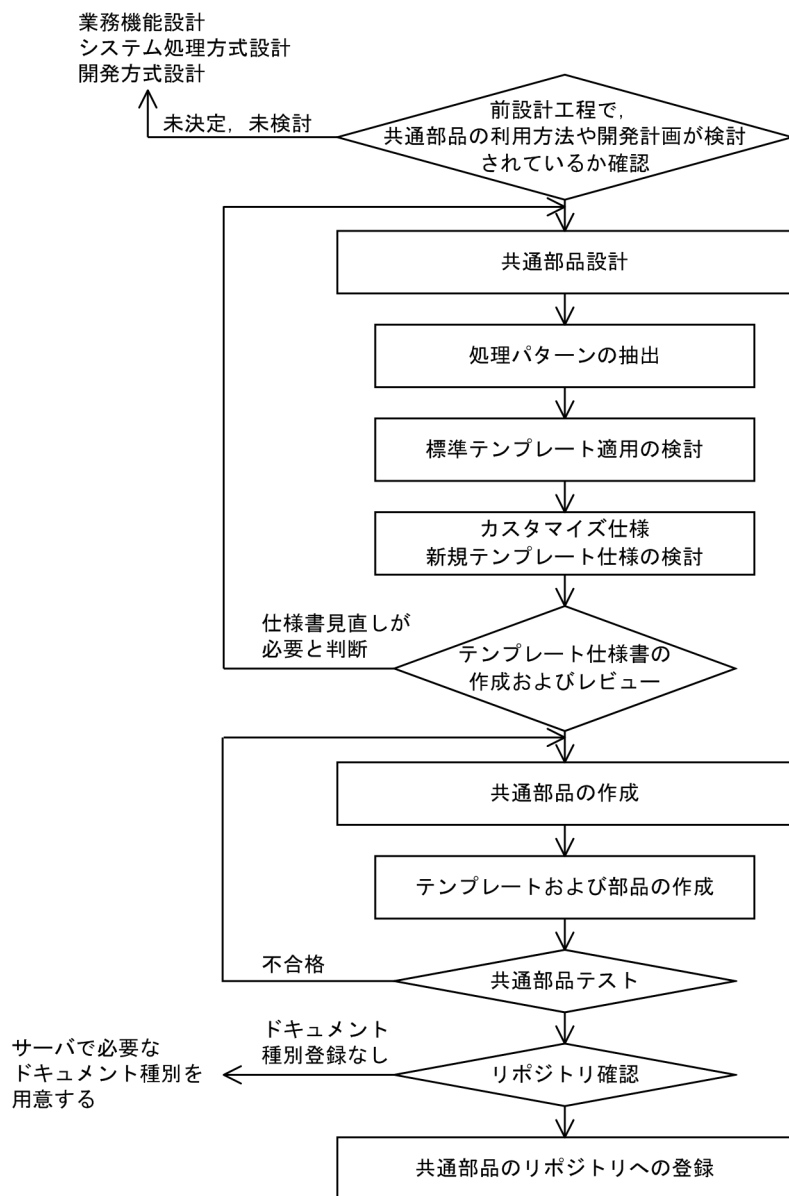
6.1 テンプレートと部品の作成

6.1 テンプレートと部品の作成

SEWB+ で利用するテンプレートおよび部品は、各 AP に共通するパターンや処理を基にして作成します。これらは、前工程で作成されている、業務機能設計書、システム処理方式設計書および開発方式設計書から抽出されたものです。

テンプレートおよび部品作成の流れを次の図に示します。

図 6-1 テンプレートおよび部品作成の流れ



6.1.1 共通部品の設計

共通部品の設計手順について説明します。

(1) 処理パターンの抽出

システム処理方式で決定した機能分散の方式をプログラム単位に分割して、各プログラムの処理を明確にします。

(2) 標準テンプレート適用の検討

(1)で抽出した処理パターンに、標準テンプレートを適用するかどうかを検討します。標準テンプレートとは、他プロジェクトで作成したものおよび SEWB+/CS-FRAMEWORK, SEWB+/BATCH-FRAMEWORK などを指します。検討後、処理パターンを次のように分類しておきます。

- 標準テンプレートをそのまま使用するもの。
- 標準テンプレートをカスタマイズして使用するもの。
- 新規にテンプレートを作成するもの。

(3) カスタマイズ仕様および新規テンプレート仕様の検討

処理パターンを分類したあと、「標準テンプレートをカスタマイズして使用するもの」、「新規にテンプレートを作成するもの」について、テンプレート部分、部品部分およびユーザ追加処理部分の区別を明確にします。それぞれの部分に含まれる内容を次に示します。

テンプレート：

- プログラムの骨格として、生成後、変更する必要のない処理。
- 入出力項目などプログラム定義のユーザインタフェースとなる項目、およびその設定時の上限値やチェック内容。設定されたデータ項目に対する展開ロジックの仕様。
- 部品、業務ルールおよびユーザ追加処理を展開する位置。

部品：

- 複数の処理パターンで共通に使用できる処理。ファイル入出力、DB アクセス、トランザクション制御などが該当する。

ユーザ追加処理：

- AP ごとに固有な処理。

(4) テンプレート仕様書の作成およびレビュー

共通部品設計での結果をドキュメントにまとめ、AP 開発者とレビューをします。ドキュメントとしてまとめる項目を次に示します。

- テンプレートから生成されるプログラムの内容、および AP ごとに固有な処理としてユーザが追加処理しなければならない処理の内容と、プログラム中での位置。
- テンプレート中のインタフェース定義部での定義内容、および定義内容と生成されるソースプログラムの関係。
- テンプレートから利用できる部品の一覧。

作成するドキュメント（仕様書）を図 6-2 ~ 図 6-7 に示します。なお、それぞれの図の仕様書およびテンプレートは、SEWB+/CONSTRUCTION のサンプルとして提供されています。

6. 適用テンプレート・部品の準備

図 6-2 パターンテンプレート仕様書 (1)

パターンテンプレート仕様書				名称	パッチパターン (DB→SAM) 2.cst パッチテンプレート (DB→SAM)	作成	生産技術部	承認	P. 1
				処理概要 DBを読み込み、変換後、 出力ファイルに出力する。					
merge記号			global変数						
No	名称	位置 (section)	展開内容	No	名称	値	用途		
1	@FILE_SELECT	INPUT-OUTPUT	指定したファイルのSELECT句	1	@ErrFlg	入力パラメータエラー時 1を既定	値が1の時、@errorexit		
2	@FILE_FD	FILE	指定したファイルのFD句	2	@InCnt	@入力ファイルの指定 数	入力用作業領域の展開有無の決定に使用		
3	@prog_working	WORKING-STORAGE	展開する部品中で使用するワーク エリア	3	@OutCnt	@出力ファイルの指定 数	出力用作業領域の展開有無の決定に使用		
4	@count_working	WORKING-STORAGE	開始・終了、入出力件数出力用ワ ークエリア	4	@FileCnt	@InCnt*@OutCnt	OPEN句、CLOSE句の展開に使用		
5	@sql_whenver	MAIN-PROC	SQLERROR、NOT FOUND用TRIGGERS	5	@DBCnt	@DBファイルの指定 数	埋め込み変数の展開に使用		
6	@start_msg	INIT-PROC	プログラム開始メッセージ出力処 理	6	@WrkCnt	OPEN句、CLOSE句の展 開カウンタ	OPEN句、CLOSE句の展開に使用		
7	@count_init	INIT-PROC		7	@WRK_IDCnt	埋め込み変数の展開カ ウンタ	埋め込み変数の展開に使用		
8	@db_connect	INIT-PROC	DBへのCONNECT処理						
9	@db_commit	END-PROC	DBへのCOMMIT処理						
10	@count_msg	END-PROC	入出力件数メッセージ出力処理						
11	@end_msg	END-PROC	プログラム終了メッセージ出力処 理						

図 6-3 パターンテンプレート仕様書 (2)

パターンテンプレート仕様書				名称	パッチパターン (DB→SAM) 2.cst パッチテンプレート (DB→SAM)	作成	生産技術部	承認	P. 2
インタフェース定義									
No	名称	種別 (識別子)	戻値	I/O	REF	コメント			
1	@入力ファイル	FILE	5	IN		変換元の入力ファイルを指定する。			
	ファイル名	FILE_NAME							
	ファイル所属	ORGANIZATION							
	レコード名	RECORD_NAME							
	コピーメンバ名					コピー句を使用する場合のメンバ名を入力する。			
	外部装置名					外部装置名を入力する。			
2	@出力ファイル	FILE	5	OUT		変換先の出カファイル指定する。 (1件目をメインの出カファイルとする。 最後1件入力必須)			
	ファイル名	FILE_NAME							
	ファイル所属	ORGANIZATION							
	レコード名	RECORD_NAME							
	コピーメンバ名					コピー句を使用する場合のメンバ名を入力する。			
	外部装置名					外部装置名を入力する。			
3	@入力DBファイル	DB		IN		入力となるDBファイルを指定する。			
	テーブル名	TABLE_NAME							
	レコード名	RECORD_NAME							
	接続語					接続語を入力する。			
	接続語					接続語を入力する。			
	接続語					接続語を入力する。			
4	@DBファイル	DB	15	IN_OUT		参照、更新するDBファイルを指定する。			
	テーブル名	TABLE_NAME							
	レコード名	RECORD_NAME							
	接続語					接続語を入力する。			
	接続語					接続語を入力する。			
	アクセスモード					A~Eのいずれかを入力する。 A:一件検索 B:一件検索、追加、更新、削除 C:複数検索 D:複数検索、追加、更新、削除 E:追加			
5	@WORKテーブル	WORK				ワーク用テーブルを指定する。 DBやファイルからの入出力データは、全てこのテーブルで管理する。			

図 6-4 パターンテンプレート仕様書 (3)

パターンテンプレート仕様書		名 称	作成	承認	生産技術部	P. 4
機能説明		バッチパターン (DB→SAM) 2.cst				
処理構成		バッチテンプレート (DB→SAM)				
		コメント				
<input type="checkbox"/> ユーザーワークエリア >メイン処理		作業開始に追加する変数を記述する。 1. 初期処理を行う。 2. @入力DB ファイルカーソル宣言&OPEN 3. @入力DB ファイルを読む 4. @入力DB ファイルレコードがなくなるまで、以下の処理を行う。 4.1 DBレコードチェック処理 4.2 出力フラグに1が設定されている時、以下の処理を行う。 4.2.1 出力ファイルレコード編集処理 4.2.2 出力ファイル[1]レコードを書き込む 4.3 @入力DB ファイルを読む 5. @入力DB ファイルカーソルCLOSE 6. 終了処理を行う。				
>初期処理 >初期処理_UOC		プログラム開始メッセージ出力処理 ファイルOPEN処理 DBへのCONNECT処理【初期処理_UOC】 初期処理に追加する処理を記述する。				
>DBレコードチェック処理 >DBレコードチェック処理_UOC		[DBレコードチェック処理_UOC] DBレコードチェック処理を記述する。 ファイルに出力したい場合は、DB-OUTE-FLAGに'1'をセットする。				
>出力ファイル編集処理 >出力ファイル編集処理_UOC		[出力ファイル編集処理_UOC] 出力ファイル編集処理を記述する。 ファイルに出力したい値を、@file[レコード名]にセットする。				
>終了処理 >終了処理_UOC		DBへのCOMMIT処理 入出力件数メッセージ出力処理 ファイルCLOSE処理 プログラム終了メッセージ出力処理 [終了処理_UOC] 終了処理に追加する処理を記述する。				

図 6-5 パターンテンプレート仕様書 (4)

パターンテンプレート仕様書				名 称	作成	承認	生産技術部	P. 5
使用部品				業務ルール展開				
No	ファイル名	プロシジャ名	パラメタ	No	展開名	i/o	インタフェース	
1	ファイルアクセス	file_check	@入力ファイル					
2	ファイルアクセス	file_check	@出力ファイル					
3	DBアクセス	file_check	@入力DBファイル					
			''					
			''					
4	DBアクセス	file_check	@DBファイル					
			'd'					
			'e'					
5	ファイルアクセス	file_open	@入力ファイル					
			'i'					
6	ファイルアクセス	file_open	@出力ファイル					
			'o'					
7	ファイルアクセス	file_close	@入力ファイル					
8	ファイルアクセス	file_close	@出力ファイル					
9	ログメッセージ編集	prog_msg	@プログラムID					
10	ログメッセージ編集	file_msg	@入力ファイル					
11	ログメッセージ編集	file_msg	@出力ファイル					
12	ファイルアクセス	file_read	@入力ファイル					
			@WORKテーブル					
13	ファイルアクセス	file_write	@出力ファイル					
			@WORKテーブル					
14	DBアクセス	DB_CONNECT	@DBユーザID					
			@DBパスワード					
15	DBアクセス	DB_COMMIT						
16	ログメッセージ編集	db_msg	@入力DBファイル					
			{ 2 }					
17	DBアクセス	DB_CURS	@入力DBファイル					
			"NOLOCK"					
18	DBアクセス	DB_EDIT	@入力DBファイル					
			"1"					
			@WORKテーブル					
19	ログメッセージ編集	db_msg	@DB					
			{ 1 }					
20	DBアクセス	DB_SLCT	@DB					
			"NOLOCK"					

6. 適用テンプレート・部品の準備

図 6-6 部品テンプレート仕様書 (1)

部品テンプレート仕様書			名 称	DBアクセス、csr DB入出力部品	作成 承認	生産技術部	P. 1
処 理 概 要			No	名称	パラメタ	パラメタの意味	
DBアクセスに必要な権限及び処理を 展開する。(H I R D B用)					@curs	削除レコードの指定 1の時、WHERE句で条件を指 定 1以外の時、CURRENT カーソル	
			7	DB_COMMIT			
			8	DB_CONNECT	@usr @pwd	DBに接続するユーザID DBに接続するユーザIDのパス ワード	
			9	DB_ERR			
			10	DB_EDIT	DB_info	@interfaceで指定した可変記号 (ATT=ID,ARRAY_MAX=N)	
プロシジャー一覧			No	名称	パラメタ	パラメタの意味	
1	file_check	@DB_info @a_cs_s @a_cs_e			@DB_info @lock_ref	@interfaceで指定した可変記号 (ATT=ID,ARRAY_MAX=N) アクセスモードの範囲 (最小値) アクセスモードの範囲 (最大値)	
2	DB_SUCT	@DB_info @lock				@interfaceで指定した可変記号 (ATT=ID,ARRAY_MAX=N) 排他制御の指定 LOCK : 排他する LOCK以外: 排他しない	
3	DB_CURS	@DB_info @lock				@interfaceで指定した可変記号 (ATT=ID,ARRAY_MAX=N) 排他制御の指定 LOCK : 更新排他する LOCK以外: 排他しない	
4	DB_INST	@DB_info				@interfaceで指定した可変記号 (ATT=ID,ARRAY_MAX=N)	
5	DB_UPDT	@DB_info @curs				@interfaceで指定した可変記号 (ATT=ID,ARRAY_MAX=N) 更新レコードの指定 1の時、WHERE句で条件を指 定 1以外の時、CURRENT カーソル	
6	DB_DEL	@DB_info				@interfaceで指定した可変記号 (ATT=ID,ARRAY_MAX=N)	

図 6-7 部品テンプレート仕様書 (2)

部品テンプレート仕様書			名 称	DBアクセス、csr DB入出力部品	作成 承認	生産技術部	P. 2
プロシジャー名	file_check		mergeへの出力				
処理内容	処理構成	コメント	No	名称	コメント		
file_check		DB権限のチェック (テーブル名、レコード名→必須) (アクセスモード→パラメタで設定した範囲内) DBアクセス用埋め込み変数の展開	1	@prog_working	BEGIN DECLARE と CONNECT用ユーザID、パ スワード変数		
			2	@prog_working	DBアクセス用埋め込み変 数		
			3	@prog_working	DB単位に展開 END DECLARE		
使用部品	業務ルール展開						
No	ファイル名	プロシジャー名	パラメタ	No	展開名	i/o	インタフェース

6.1.2 共通部品の作成

共通部品の作成手順について説明します。

(1) テンプレートおよび部品の作成

共通部品設計で作成したテンプレート仕様書を基に、テンプレートおよび部品を作成します。

(2) 共通部品テスト

作成したテンプレートおよび部品を使用して、実際にプログラムを生成するテストをします。次に示す項目に重点を置いてテストしてください。

- インタフェース定義部の定義項目の妥当性。
項目の過不足，配列の数などをチェックします。
- インタフェース定義部の定義内容に対するチェック処理の妥当性。
- SEWB+/CONSTRUCTION のプログラム定義の [ユーザ処理] タブに表示されるセクション，およびユーザ処理の階層構造の妥当性。
なお，ユーザ処理とは「業務ルール」，「部品」および「ユーザ追加処理」を指します。
- プログラム定義時に，AP 開発者に対して表示されるコメントの妥当性。
- 生成されたソースプログラムの内容。
意図した内容になっているかどうかをチェックします。
- ユーザ処理の位置の妥当性。
- 生成されたプログラムの動作確認（実機確認）。

(3) リポジトリ確認

共通部品を管理するリポジトリに，必要なドキュメント種別が登録されているかどうかを確認します。登録されていない場合は，サーバで登録してください。

(4) 共通部品のリポジトリへの登録

作成した共通部品をリポジトリに登録します。

7

サーバ AP の設計

業務機能設計やユーザインタフェース基本設計で作成したワークシートを基に、業務処理の流れやプログラム単位の処理、および入出力データの関連を明確にします。

7.1 TP モニタを使用したサーバ AP の設計

7.2 帳票出力する AP の設計

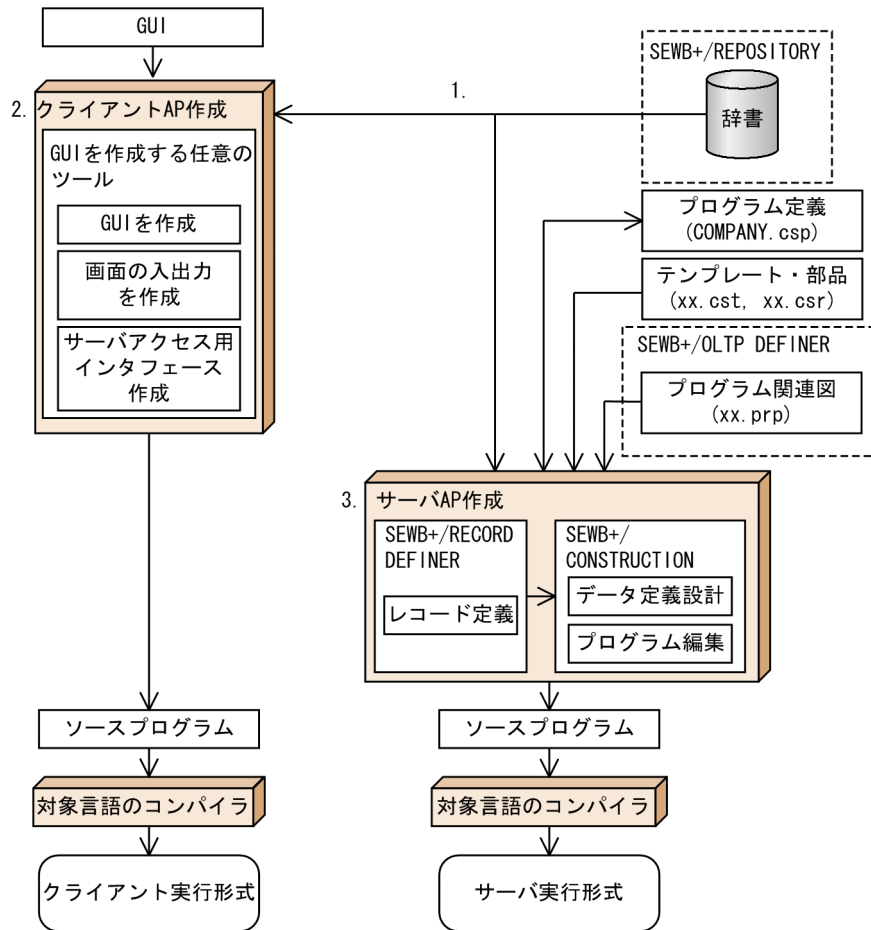
7.1 TP モニタを使用したサーバ AP の設計

TP モニタを使用したサーバ AP の設計方法について説明します。

7.1.1 TP モニタ使用時のツールの関連

TP モニタを使用した C/S システムのサーバ AP を設計する場合に利用するツールと、それぞれの関連について次の図に示します。

図 7-1 TP モニタ使用時のツールの関連



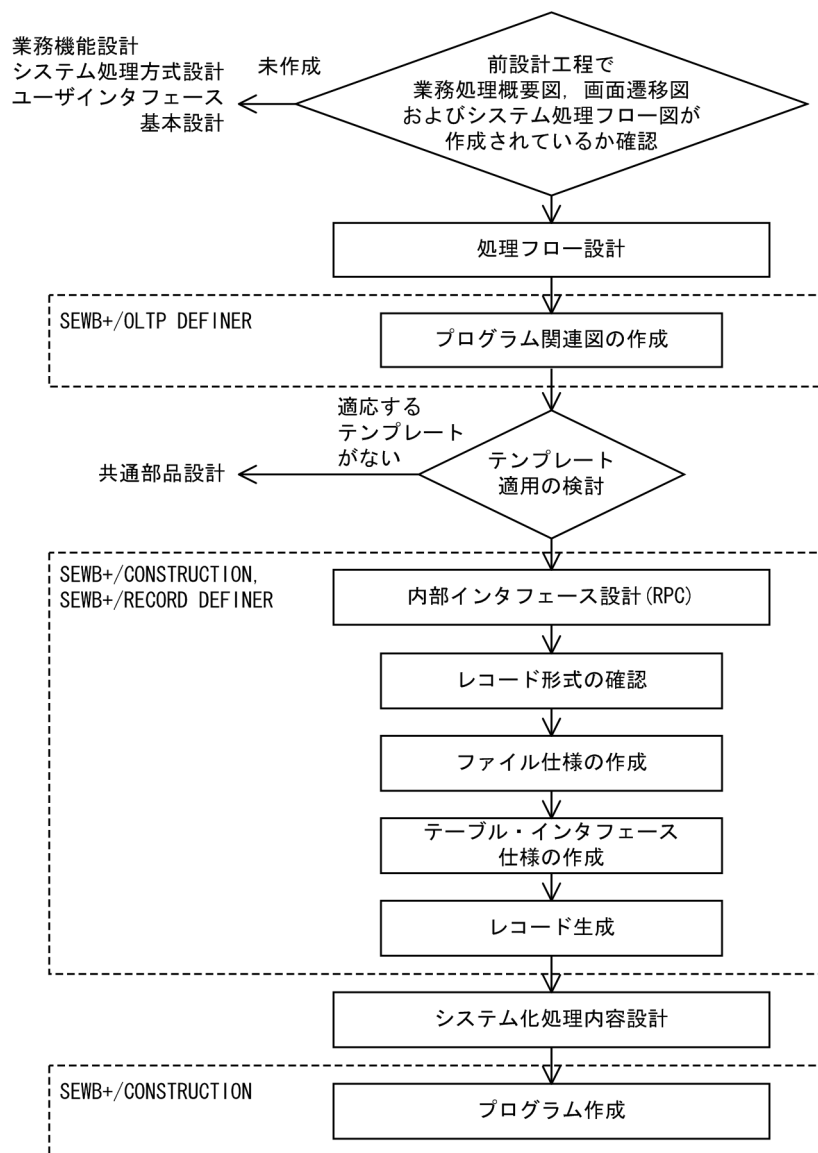
1. クライアント AP 作成に必要なデータ項目、およびサーバ AP 作成に必要なデータ項目を SEWB+/REPOSITORY の辞書から参照する。
2. GUI を作成する任意のツールで、クライアント AP を作成する。
3. SEWB+/CONSTRUCTION でサーバ AP を作成する。
SEWB+/CONSTRUCTION では、クライアント AP 用のレコードソースを生成します。これは、クライアント側の AP がサーバにアクセスするためのインターフェース部分になるため、クライアント側のソースファイル中に格納されます。

7.1.2 サーバ AP 作成の流れ

TP モニタを使用した C/S システムのサーバ AP を作成する場合、システム化機能の詳細設計をします。システム化機能詳細設計では、処理フロー設計、内部インタフェース設計およびシステム化処理内容設計が必要です。ただし、これらを設計する場合、業務機能設計で業務処理概要図が作成されていること、およびユーザインタフェース基本設計で画面遷移図が作成されていることを確認し、作成されていない場合は、該当する工程に戻って作成してください。

TP モニタを使用した C/S システムでサーバ AP を作成する手順を次の図に示します。

図 7-2 TP モニタを使用した C/S システムでサーバ AP を作成する手順



(1) 処理フロー設計

処理フロー設計では、ユーザインタフェース基本設計で作成した画面遷移図とシステム処理方式設計で作成したシステム処理フローから、プログラム関連図を作成します。プログラム関連図では、オンライン処理のプログラム間および DB 間の関連を明確にします。なお、処理フロー設計では、適用するテンプレートを考慮し、プログラム関連図を追加したり修正したりします。

(a) レコード形式の確認

設定するレコード形式は、すでに「最上位結合項目」としてデータ項目辞書に登録されているか、または SEWB+/RECORD DEFINER で定義されている必要があります。必要な最上位結合項目またはレコード定義ファイルがない場合は、データ項目管理者やシステム設計者に依頼して作成してください。

(b) ファイル仕様の作成

ファイルのレコード形式を設定します。なお、ファイルは ER 図のエンティティと対応します。そのため、ファイルに設定するレコード項目（最上位結合項目）は、ER 図のエンティティを取り込んで作成された結合項目、またはレコード定義ファイルから選ばれます。

(c) テーブル仕様およびインタフェース仕様の作成

RPC インタフェースなどのプログラム間インタフェースや、主記憶上の内部テーブルについて、それぞれ仕様書を作成します。テーブル仕様およびインタフェース仕様には、テーブルなどの具体的な構成情報を含む結合項目またはレコード定義ファイルを設定します。

(d) テーブル仕様およびファイル仕様からのレコードソースの生成

コンパイル時にレコードソースを COPY 文で取り込む場合は、SEWB+/CONSTRUCTION のデータ定義機能、または SEWB+/RECORD DEFINER のレコード定義機能でレコードソースを生成しておきます。なお、テンプレート中に「@@expand 文」を記述すると、展開時、レコード情報を自動的に取り込んでレコードソースを生成します。そのため、データ定義機能またはレコード定義機能でレコードソースを生成しておく必要はありません。

(3) システム化処理内容設計

プログラム関連図に記述した各プログラムについて、入出力と処理の概要をまとめ、プログラム処理概要図を作成します。システム化処理内容設計書は、プログラム仕様設計で使用されます。

プログラム処理概要図を次の図に示します。

図 7-4 プログラム処理概要図

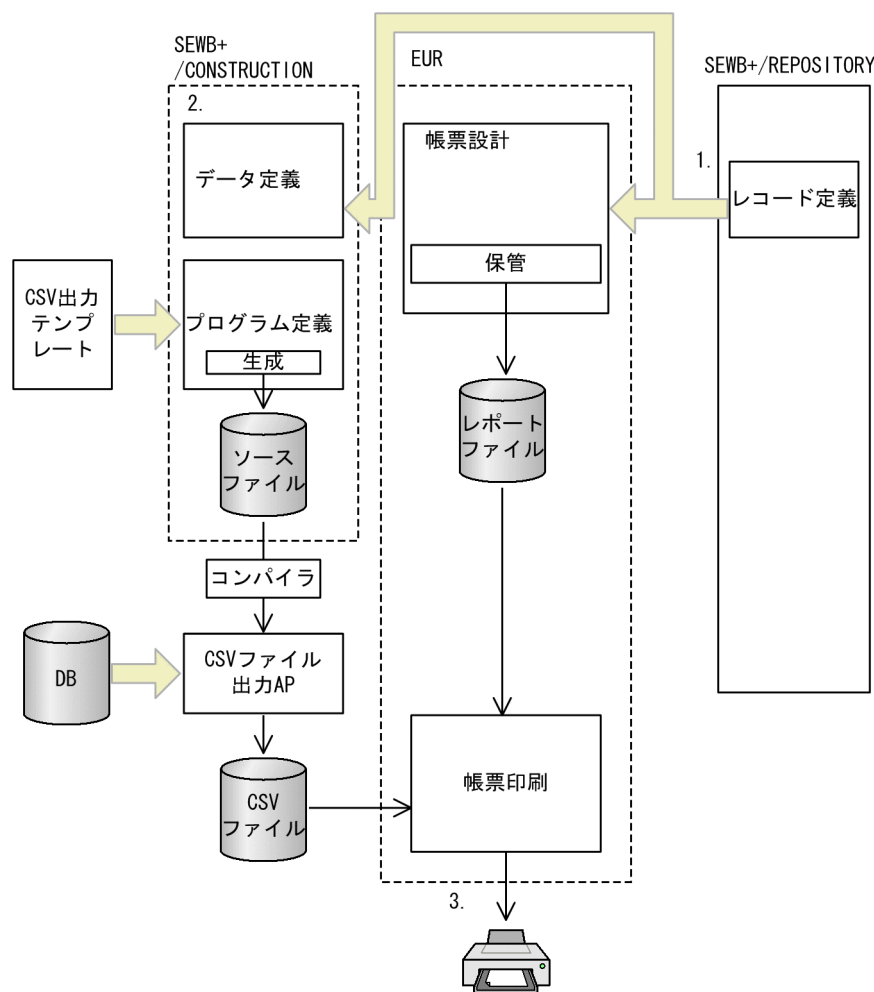
ワークシート		プログラム処理概要図		作成	日立 本部	印刷日付	P. 1
				作成		1994年09月9日	
サブシステム名		プログラム名		アプレット名	書種	テンプレート名	ページ数
出金管理		旅費集計		TRAVEL01	CONTROL	レコードの集計	1
<p>処理系図</p>				<p>概要</p> <ol style="list-style-type: none"> 旧トランザクションファイルとマスタファイルを読み込み、照合を行い、新トランザクションファイルに出力する。 旧トランザクションキーがマスタファイルがマスタファイルキーより小さいとき、そのトランザクションレコードをエラーとして、エラーファイルに出力する。 旧トランザクションファイルの読み込みが終了した時点で処理を終了する。 			
入出力定義							
番号	入出力項目	種別	IO	ファイル名	記号-句	共有変数 (Y/N/R)	検索語
1	入力ファイル	FILE	I	旅費申請情報ファイル	MS TRIP	010010	S
2	出力ファイル	FILE	O	旅費申請情報ファイル	MS TRIP	010020	S
パラメタ							
番号	表示項目	値	説明				
1	プログラム種別		プログラムの種別を記述する。				
2	プログラムID		プログラムIDを記述する。				
3	プログラム種別フラグ		'R'または'G'を入力する。				

7.2 帳票出力する AP の設計

EUR は、SEWB+/REPOSITORY の資源を利用して、帳票出力用 AP を作成するツールです。EUR では、帳票に使用する辞書のレコード定義（最上位結合項目）を選択し、帳票を構成するフィールドの定義情報を生成します。生成された情報を基に、レイアウトを編集し、帳票を設計します。帳票を印刷する場合は、設計した帳票と、SEWB+/CONSTRUCTION で作成した実行ファイルから出力されるデータファイルを読み込んで、印刷します。

帳票出力する AP を設計するツールと AP の関係を次の図に示します。

図 7-5 帳票出力する AP を設計するツールと AP の関係



1. EUR で帳票を出力する AP を作成するために、リポジトリからレコード定義（最上位結合項目）を参照する。
2. SEWB+/CONSTRUCTION で CSV ファイルを出力する AP を作成するために、リポジトリからレコード定義（最上位結合項目）を参照する。
3. EUR が設計した帳票と、SEWB+/CONSTRUCTION が作成した実行ファイルから出力されるデータファイルを読み込んで、印刷する。

8

プログラムの作成

システム化機能詳細設計の結果を基に，ソースプログラム作成に使用するテンプレートや部品を指定してプログラム定義をし，ソースプログラムを生成します。また，生成されたソースプログラムをコンパイル・リンケージして実行ファイルを作成します。プログラム定義およびプログラム作成については，マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

8.1 プログラム定義の準備

8.2 プログラムの作成（TP モニタ）

8.3 ソースプログラムの生成と修正手順

8.1 プログラム定義の準備

これまでに作成したテンプレートや部品をリポジトリに登録し、SEWB+/CONSTRUCTION の環境設定でパスを設定しておきます。

8.1.1 SEWB+/CONSTRUCTION での環境設定

プログラム定義の前に、SEWB+/CONSTRUCTION の環境設定で設定しておく項目を次に示します。詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

- 生成オプション
ソースプログラム自動清書の抑止やインデントサイズなどを設定します。
- パス
プログラム定義で使用するファイルの検索パスを指定します。
- エディタ
ユーザ追加処理や生成時のソースプログラム表示に使用するエディタを設定します。任意のエディタを設定できます。デフォルトは「Notepad.exe」です。
- 業務ルール
すべての業務ルールを無条件に展開するかどうかを設定します。
- ユーザ追加処理
ユーザ追加処理の開始および終了を示す文字列などを設定します。
- ファイル
プログラム定義ファイルおよびデータ定義ファイルのファイル形式を指定します。バイナリ形式、または XML 形式を指定できます。

8.1.2 パス設定

リポジトリに登録したテンプレートや部品へのパスを設定する手順について説明します。

(1) ネットワーク参照のためのパス設定

SEWB+/CONSTRUCTION の環境設定パスでは、「テンプレート」、「データ定義」、「部品」、「XML」、「マップ定義」および「レコード定義」のファイルのパスを設定します。これらのファイルは、ルートドキュメントフォルダ下に格納されています。ルートドキュメントフォルダは OS で共有宣言されているため、チェックアウトしなくてもローカル環境のマシンからネットワーク参照できます。ネットワーク参照する場合は、環境設定のパス設定でネットワークパスを設定してください。

(2) パスの設定方法

プログラム定義で使用するファイルのパスの設定方法を示します。

(a) 相対パスでの設定

SEWB+/CONSTRUCTION の環境設定でパスを設定しておく、プログラム定義で使ったファイルはファイル名（相対パス）で保存されます。パスは、あらかじめ環境設定で設定しておくことをお勧めします。

(b) 絶対パスでの設定

SEWB+/CONSTRUCTION の環境設定でパスを設定していない場合、プログラム定義で使ったファイルは、ドライブ名やコンピュータ名を含むファイル名（絶対パス）で保存されます。

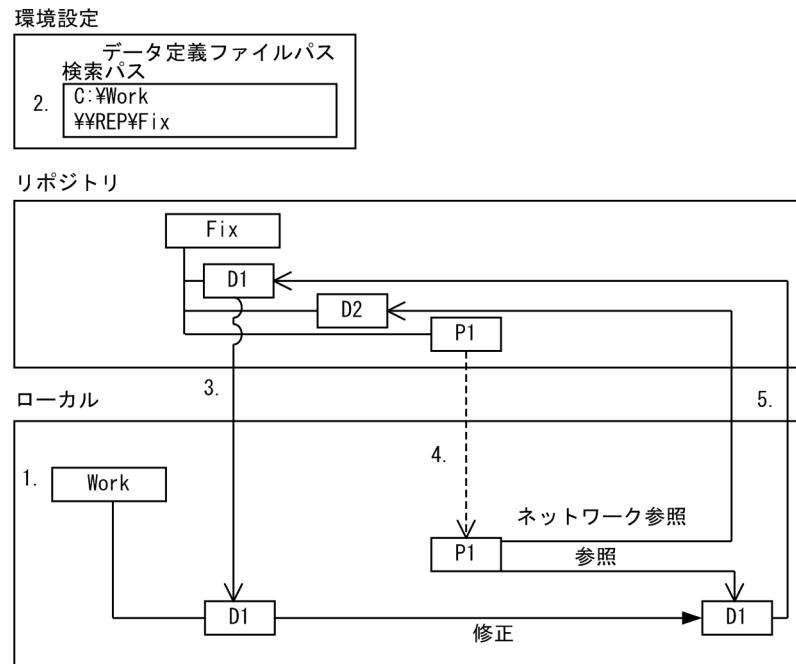
また、絶対パスを相対パスに変換できます。絶対パスから相対パスへの変換については、マニュアル

「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

(c) 複数パスの設定

環境設定では、複数のパスを設定できます。例えば、テンプレートファイルを修正して確認するために、修正前または修正中のファイルと修正後のファイルを別々のパスで保管します。それぞれのパスを検索したい順番に従って環境設定に設定しておくこと、設定した順番にプログラム定義で使用されるので便利です。複数パスの設定方法については、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。複数パスの設定を利用した作業手順を次の図に示します。

図 8-1 複数パスの設定を利用した作業手順



- データ定義を修正するために、修正作業用のフォルダ「Work」をローカルに作る。
このとき、リポジトリには次のドキュメントが格納されていると仮定します。これらはリポジトリのフォルダ「Fix」に格納されています。
 - D1：修正したいデータ定義ファイル
 - D2：修正しないデータ定義ファイル
 - P1：D1，D2を参照するプログラム
- 環境設定で「Work」，「Fix」の順番で複数パスを設定しておく。
「Work」はローカルパスで、「Fix」はネットワークパスでそれぞれ設定します。修正作業用のフォルダを必ず最初に設定してください。
- リポジトリから修正したいデータ定義を「Work」にチェックアウトして、修正する。
- 修正が終わったら、修正したデータ定義の妥当性をテストするために、プログラム定義を起動させる。
複数パスの設定で先に「Work」が設定されているため、必ず「Work」のデータ定義が参照されます。「P1」が参照する「D1」は修正された「Work」フォルダのものが参照され、「D2」はリポジトリの「Fix」フォルダのものがネットワーク参照されます。
- テストでデータ定義の妥当性が確認されたら、「Work」のデータ定義を「Fix」にチェックインします。
以降、プログラム定義を起動すると、複数パスの最初に設定された「Work」には該当するデータ定義

がないため、2番目に設定された「Fix」のデータ定義を参照するようになります。

注

「Work」のデータ定義を「Fix」にチェックインするときには、ローカルファイルに不要な資源を残さないように「チェックアウトファイルを削除する」が選ばれていることを確認し、選ばれていなければ選んでください。

(d) 複数パス設定の注意事項

環境設定で複数のパスを設定した場合、先に設定されたパスが優先されます。例えば、「C:¥ 修正中データ定義」、「¥¥NET¥ データ定義」の順番に設定されている場合、「¥¥NET¥ データ定義」のファイルを参照したい場合でも、同じ名前のファイルが「C:¥ 修正中データ定義」にあれば、そちらのファイルが参照されます。

8.2 プログラムの作成 (TP モニタ)

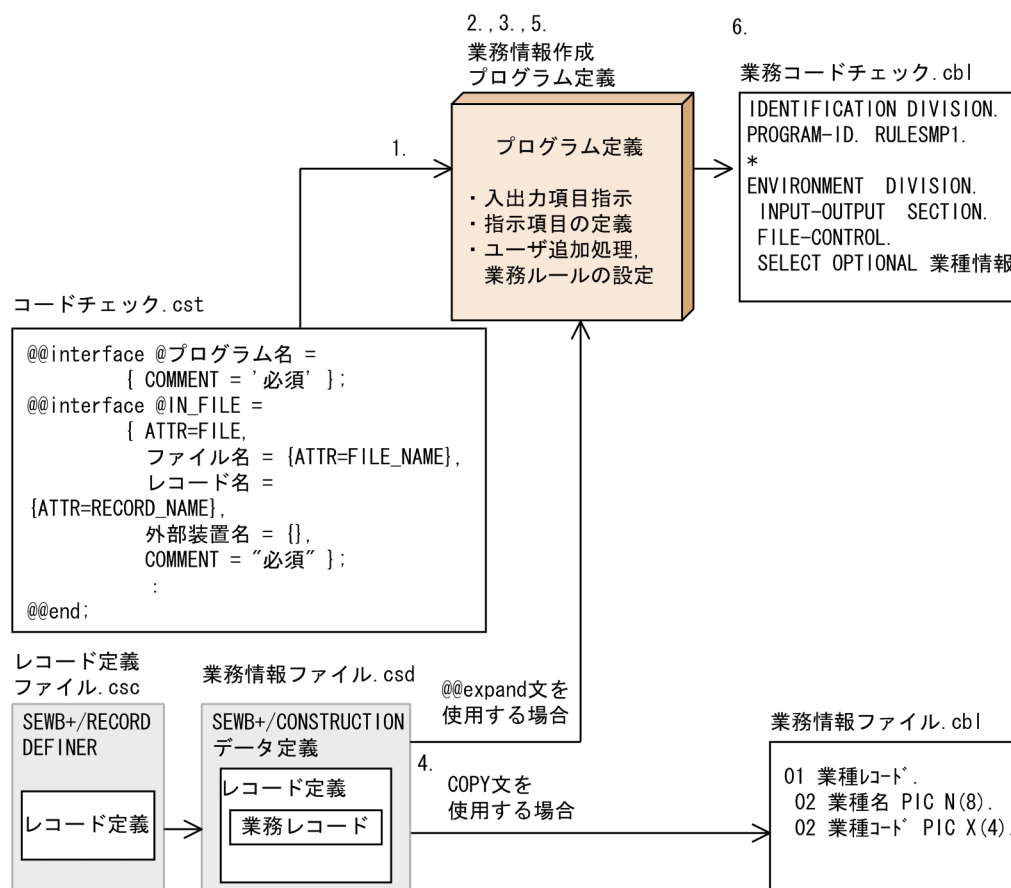
SEWB+/CONSTRUCTION で、TP モニタを利用した C/S システム上で動作する AP を作成する手順について説明します。

8.2.1 TP モニタで動作する AP の作成

TP モニタで動作する AP の作成例について説明します。この例では、コードチェックプログラム作成用のテンプレートファイル「コードチェック .cst」をプログラム定義で選択します。そのあと、データ定義を選択して入出力項目を指示し、業務コードチェックのソースプログラムを生成します。データ定義ファイル「業務情報ファイル .csd」からは、レコードソースを生成します。

SEWB+/CONSTRUCTION のプログラム定義では、ソースプログラム生成に必要な情報を各タブで定義します。ここでは、TP モニタ利用時のプログラム定義で使用するタブと、定義内容を説明します。詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

図 8-2 TP モニタで動作するプログラムの作成



作成手順

1. テンプレート選択 ([サイン] タブ)

目的の AP を作成するためのテンプレートを選びます。

AP が提供するサービスの内容は、選んだテンプレートで決まります。

なお、選んだテンプレートは変更できます。ただし、その際は、同一の可変記号だけが引き継がれま

す。

2. 入出力項目指示 ([入出力] タブ)

選択したテンプレートが AP 内で DB をアクセスする場合は DB の定義情報を、メッセージを送受信する場合はメッセージの定義情報を参照するので、それらが定義されているデータ定義ファイルを指定します。

3. 指示項目の定義 ([パラメタ] タブ)

テンプレートでの宣言に従って、プログラム定義で指示項目に値を設定します。なお、途中でテンプレートを変更した場合、同一の変記号だけが引き継がれます。

4. レコードの生成

COPY 文を使用して AP を作成する場合、データ定義ファイルからレコードソースを生成します。

@@expand 文を使用する場合は、展開時に自動的にレコード情報を取り込んでレコードソースを生成するので、データ定義ファイルからレコードソースを生成しておく必要はありません。

なお、レコードソースは、レコード定義ファイルからも生成できます。詳細は、マニュアル「SEWB+/RECORD DEFINER 操作ガイド」を参照してください。

5. ユーザ追加処理・業務ルールの設定 ([ユーザ処理] タブ)

[ユーザ処理] タブで、ユーザ追加処理として AP 固有の処理を編集します。また、AP で使用する業務ルール処理を設定します。同一項目用業務ルールを使用すると、ユーザ追加処理の記述量を軽減できます。

6. ソースプログラム生成

プログラム定義で定義した情報に基づき、ソースプログラムを生成します。

8.2.2 プログラム作成時の注意事項 (TP モニタ)

ソースプログラム中でユーザが追加するプログラム固有の処理 (ユーザ追加処理) の編集、およびソースプログラム中で展開する業務ルールを設定をします。

ここで定義するユーザ追加処理や業務ルール処理の内容、数および挿入位置は、[入出力] タブや [パラメタ] タブの内容で変わります。[入出力] タブや [パラメタ] タブを変更した場合は、[ユーザ処理] タブの「編成のリフォーム」で最新の情報を反映しておきます。

なお、業務ルールを取り込む場合は、プログラム定義で「辞書の利用」をチェックしておきます。

8.3 ソースプログラムの生成と修正手順

SEWB+/CONSTRUCTION でソースプログラムを新規に生成したり、生成したソースプログラムを修正したりする場合、必要なドキュメントはすべてリポジトリからチェックアウトし、作業が終わったらチェックインします。このとき、適切な手順に従ってチェックイン、チェックアウトするとドキュメント間の関連が自動的に付けられます。ドキュメント同士が適切に関連づけられていると、変更時の変更波及を調査するのに役立ちます。

8.3.1 ソースプログラムを新規に生成する場合

最初にソースプログラムを生成する場合、必要なドキュメントは SEWB+/REPOSITORY-BROWSER で新規作成します。新規作成したドキュメントはチェックアウト状態です。必要な定義をしたら、リポジトリにチェックインしてください。

8.3.2 修正時の手順

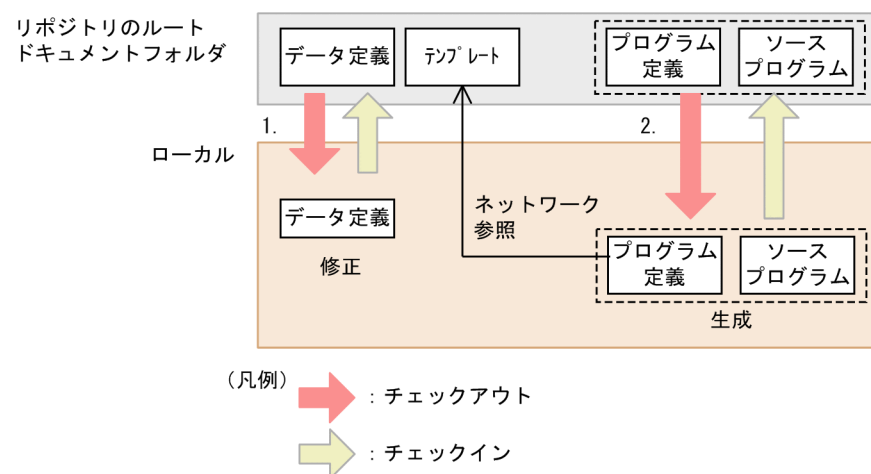
生成したソースプログラムや各種ドキュメントを修正する場合、次に示す手順に従って作業すると、ドキュメント間に適切な関連が自動的に付けられます。なお、ドキュメントを修正したい場合は、「関連」を利用して変更波及を調べてください。

(1) データ項目または業務ルールに修正が入る場合

1. SEWB+/REPOSITORY-BROWSER で、データ項目辞書または業務ルール辞書の内容を修正する。

(2) データ定義、レコード定義、テンプレート、およびマップ定義に修正が入る場合

次にデータ定義を修正する場合の例を示します。レコード定義、テンプレート、またはマップ定義を修正する場合は、データ定義をレコード定義、テンプレート、またはマップ定義に置き換えてください。



1. リポジトリから修正するドキュメントをチェックアウトして、修正したあと、チェックインする。
2. 修正したドキュメントを参照するプログラム定義とソースファイルと一緒にチェックアウトし、再生成してから一緒にチェックインする。

8.3.3 生成時および修正時の注意事項

データ定義、レコード定義、テンプレート、コンパイル時に参照される共通モジュールやテーブルなど、

8. プログラムの作成

複数 AP で共用されるファイルは、ネットワーク参照することをお勧めします。これらのファイルを、SEWB+/REPOSITORY-BROWSER でルートドキュメントフォルダに格納すると、ローカルのマシンから常に最新のリポジトリ情報を参照できます。また、リポジトリへのチェックイン時に変更波及の調査のための関連が自動設定できます。

9

コンパイルと単体テスト

コンパイラを使用して、作成したソースプログラムをコンパイルします。コンパイルが終わったら、ソースプログラムがプログラム仕様どおりに作成され、正常に動作するかどうかを確認するため、単体テストをします。

9.1 コンパイルについて

9.2 COBOL2002 でのコンパイル

9.1 コンパイルについて

システム設計で作成した設定情報を基に、テンプレート・部品の選定、展開する業務ルールやユーザ追加処理のコーディングをします。それらの作業が終わったら、ソースプログラムを生成し、コンパイルします。コンパイルには、使用している言語に対応するコンパイラを使用してください。

9.1.1 ソースプログラムのリポジトリへの登録

ソースプログラムは一般的には、ローカルフォルダで生成され、あとからリポジトリに登録されます。なお、リポジトリに登録する場合、保守などのために、ソースプログラムは設計文書と対応づけて保管します。クライアントで生成したソースプログラムの登録方法については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

9.1.2 プログラムのコンパイルとテスト

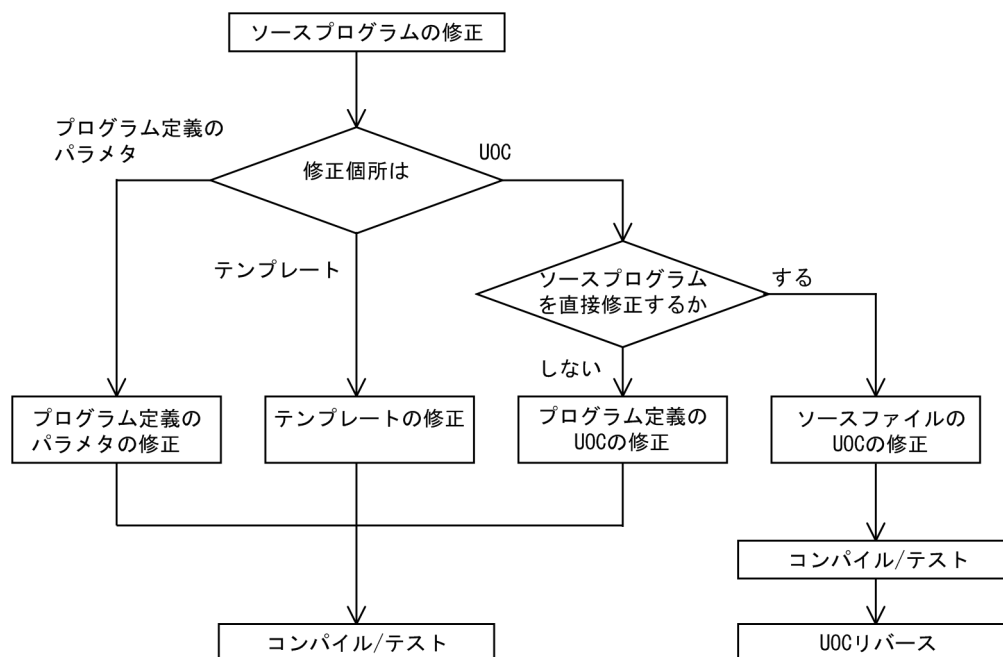
リポジトリ環境でテストをする場合、コンパイルに関連するすべてのドキュメントを参照専用チェックアウトし、ビルドを実行して、中間ファイルを含めてクライアント側のディレクトリにすべて作成します。テストを実行して問題がある場合は、修正するドキュメントを更新専用チェックアウトします。テストが終わったら、更新専用でチェックアウトしたドキュメントをチェックインし、必要な関連づけをします。ドキュメント修正時の手順については、「8.3.2 修正時の手順」を参照してください。

プログラムのコンパイルとテストの手順については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

9.1.3 プログラムの修正

プログラムのコンパイル・単体テストの際に、不良を発見したり、仕様変更が発生したりした場合は、プログラムの修正が必要になります。ソースプログラムの修正の流れを次の図に示します。

図 9-1 ソースプログラムの修正の流れ



プログラム定義のパラメタの修正

1. プログラム定義のパラメタを修正する。
2. ソースプログラムを再生成し、テストする。

テンプレートの修正

1. テンプレートを修正する。
2. プログラム定義からソースプログラムを再生成し、テストする。

注

ほかのプログラム定義でテンプレートを使用している場合があるので、SEWB+/REPOSITORY-BROWSER の関連ブラウザを使用して、必ず変更波及の調査をしてください。

プログラム定義の UOC の修正

1. プログラム定義の UOC を修正する。
2. ソースプログラムを再生成し、テストする。

ソースファイルの UOC の修正

1. ソースファイルの UOC を修正する。
2. ソースファイルを再生成し、テストする。
3. 修正した UOC をプログラム定義の UOC に取り込み、プログラム定義からソースプログラムを再生成する。

UOC をプログラム定義に取り込むには、UOC リバース機能 を使用します。

注

プログラム定義からソースプログラムを再生成したときに、取り込んだ UOC が生成されているかを確認してください。

注

UOC リバース機能を使用するには、テンプレートに UOC の開始および終了文字列の記述が必要です。UOC の開始および終了文字列の記述については、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

9.2 COBOL2002 でのコンパイル

COBOL2002 で AP を作成した場合、COBOL 開発マネージャを利用できます。ここでは、COBOL2002 の COBOL 開発マネージャを利用する場合について説明しています。なお、SEWB+/CS-FRAMEWORK および SEWB+/BATCH-FRAMEWORK を利用して AP を作成した場合も同様です。

9.2.1 COBOL 開発マネージャの利用

SEWB+/CONSTRUCTION で作成した COBOL2002 のソースファイルをコンパイルする場合、COBOL 開発マネージャを利用します。COBOL 開発マネージャの操作方法については、マニュアル「COBOL2002 操作ガイド」を参照してください。

次に、COBOL 開発マネージャの利用の概要について説明します。

(1) プロジェクトファイルの利用

COBOL 開発マネージャでソースプログラムをコンパイルする場合は、ファイルなどの開発に必要な資源を一括して管理するためのプロジェクトと、複数のプロジェクトを一括して管理するためのプロジェクトマスタを作成します。

作成方法を次に説明します。

プロジェクトマスタから新規に作成する場合

COBOL 開発マネージャの [プロジェクトマスタ] - [新規作成] を選び、プロジェクトマスタとプロジェクトを作成します。

既存のプロジェクトマスタに追加する場合

COBOL 開発マネージャの [プロジェクト] - [プロジェクトの作成] を選び、プロジェクトを作成します。

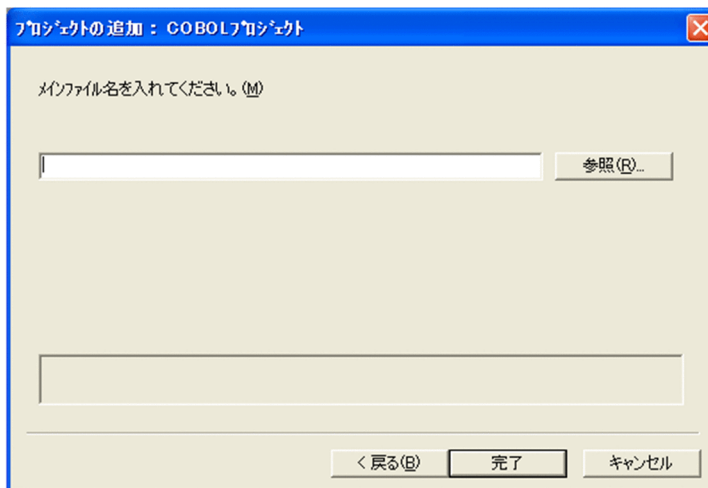
(a) ファイル関連の登録

AP 作成に使用するソースファイルおよびプログラム定義ファイルの関連を、次に説明する方法で登録します。

プロジェクトマスタから新規に作成する場合

1. COBOL 開発マネージャの [プロジェクトマスタ] - [新規作成] を選びます。

次に示すダイアログでは、メインファイル名を指定しないで [完了] ボタンをクリックしてください。



注

Windows XP の場合に表示される画面を使用しています。

2. COBOL 開発マネージャの [プロジェクト] - [ソースファイルの追加] を選び、プログラム定義ファイルを設定してください。プログラム定義ファイルとソースファイルの関連が登録されます。

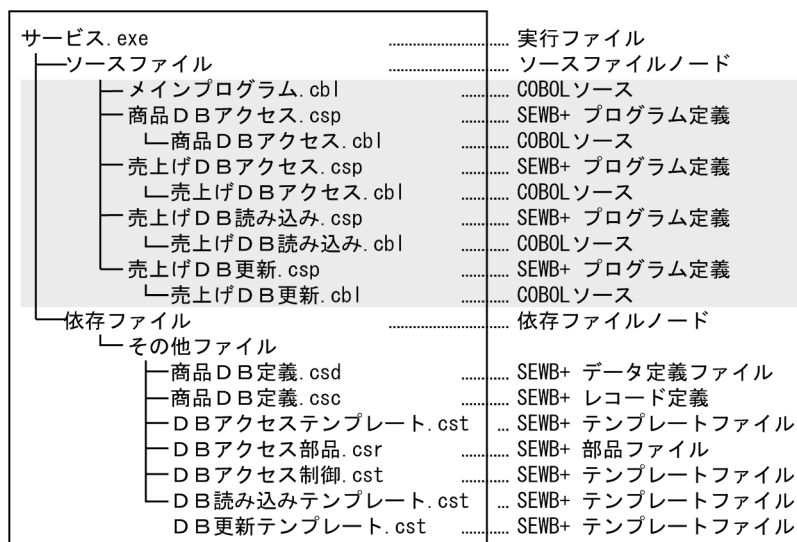
既存のプロジェクトマスタに追加する場合

COBOL 開発マネージャの [プロジェクト] - [ソースファイルの追加] を選び、プログラム定義ファイルを設定することで、プログラム定義ファイルとソースファイルの関連が登録されます。

登録したファイルをビルドすると、それぞれのソースファイルおよびプログラム定義ファイルで使用された COPY メンバ、データ定義ファイル、テンプレート、および部品の関連が付けられます。

各ファイルの関連を次の図に示します。

図 9-2 各ファイルの関連



(凡例)

■ : ユーザが登録する関連

注 この例題は、COBOLソース形式が固定形式の場合です。

(2) ビルド・リビルド機能の利用

ビルドまたはリビルドを実行すると、プロジェクトに登録された資源から実行可能ファイル (.exe, .dll) が作成されます。

データ定義やプログラム定義の修正が起こった場合は、ビルドまたはリビルド機能で自動的にソースファイルを再生成させて、実行可能ファイルを作成できます。

(3) COBOL2002 テストデバッグの利用

COBOL 開発マネージャ上でテストをする場合は、COBOL2002 のテストデバッグを起動させます。COBOL 開発マネージャ上でのテストは単体テストなので、SEWB+/CONSTRUCTION で作成したサーバ側の AP をテストする場合は、COBOL2002 のシミュレーション機能を利用してください。クライアント側の AP を COBOL2002 で作成した場合は、クライアント側の AP もテストできます。

OpenTP1 を使用しているシステムの場合は、OpenTP1 のテスト機能 を使用して、RPC およびメッセージ送受信のサーバ側の定義部分をテストできます。

単体テスト後、プロジェクトファイル、実行ファイルおよびカバレッジファイルなど作成されたファイルをリポジトリに登録します。

注

OpenTP1 をテストモードで動作させるオンラインテスタ (TP1/Online Tester), および OpenTP1 の API をテストするオフラインテスタ (TP1/Offline Tester) があります。

(4) プロジェクトマスタファイルの登録

COBOL 開発マネージャで作成したプロジェクトマスタファイルをリポジトリに登録すると、プロジェクトで定義したビルド時のファイル構成やコンパイラオプションもまとめて管理できます。

9.2.2 COBOL 開発マネージャの注意事項

COBOL 開発マネージャを使用するときは、次に示す項目に注意してください。

(1) プロジェクトマスタファイルをリポジトリに登録する場合

コンパイル・テスト後に作成されたファイルをリポジトリに登録する場合は、あらかじめ、COBOL 開発マネージャで使用するファイルをドキュメント種別としてリポジトリに登録しておきます。

(2) プリコンパイラを使用している場合 (ORACLE を利用している場合など)

最初にプリコンパイラを実行させ、プリコンパイラが生成したソースプログラムを COBOL 開発マネージャに登録します。

(3) COBOL 以外のソースがある場合

TP モニタに OpenTP1 を使用している場合、スタブソースが生成されます。スタブソースは C 言語で作成されているため、あらかじめ C 言語のコンパイラでコンパイルしておきます。

このように、COBOL 以外のソースがある場合は、あらかじめ専用のコンパイラでコンパイルし、オブジェクトファイルにしてから COBOL 開発マネージャに登録します。

(4) ソースファイルに修正が発生した場合

ソースファイルに修正が必要になった場合、「9.1.3 プログラムの修正」に示す手順でソースファイルを

修正してください。

なお、一度リポジトリに登録されたプロジェクトファイル、プログラム定義、および COBOL ソースを修正する場合は、事前にリポジトリからチェックアウトしてください。

注

ビルドを利用すると、定義情報が変更されたものだけを自動的に再生成します。

(5) リポジトリで管理されているデータ定義ファイルを利用して、ビルドまたはリビルドを行う場合

ビルドまたはリビルドを行う前に、SEWB+/REPOSITORY に接続しておく必要があります。

ただし、SEWB+/REPOSITORY-BROWSER が起動されている場合は、すでに SEWB+/REPOSITORY と接続されているため、特に作業する必要はありません。

(6) 辞書を参照してコンパイルする場合

COBOL 開発マネージャの [プロジェクト] - [プロジェクトの設定] を選び、[プロジェクトの設定] ダイアログを起動します。[プロジェクトの設定] ダイアログのプロジェクト一覧でプログラム定義ファイルを選択したあと、コンパイラオプションに「/ud」を設定してください。

「/ud」については、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」のコマンドでの生成（ソースプログラム）について記載されている個所を参照してください。

(7) プログラム定義ファイルを最新状態にする場合

COBOL 開発マネージャの [プロジェクト] - [プロジェクトの設定] を選び、[プロジェクトの設定] ダイアログを起動します。[プロジェクトの設定] ダイアログのプロジェクト一覧でプログラム定義ファイルを選択したあと、コンパイラオプションに「/r」を設定してください。

「/r」については、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」のコマンドでの生成（ソースプログラム）について記載されている個所を参照してください。

10 統合テスト

個々のツールで作成した AP 同士が，連携して動作するかどうかを確認するため，実行マシン上で統合テストをします。

10.1 統合テストの準備

10.2 ターゲットシステムでのテストと修正

10.1 統合テストの準備

統合テストは、実際にシステムを実行させるマシン上で行います。そのため、統合テストをする前に、Windows 上で開発したソースプログラムなどの生成物を、実行マシンに転送しなければなりません。

なお、このマニュアルでは、実行マシンのことをターゲットシステムといいます。

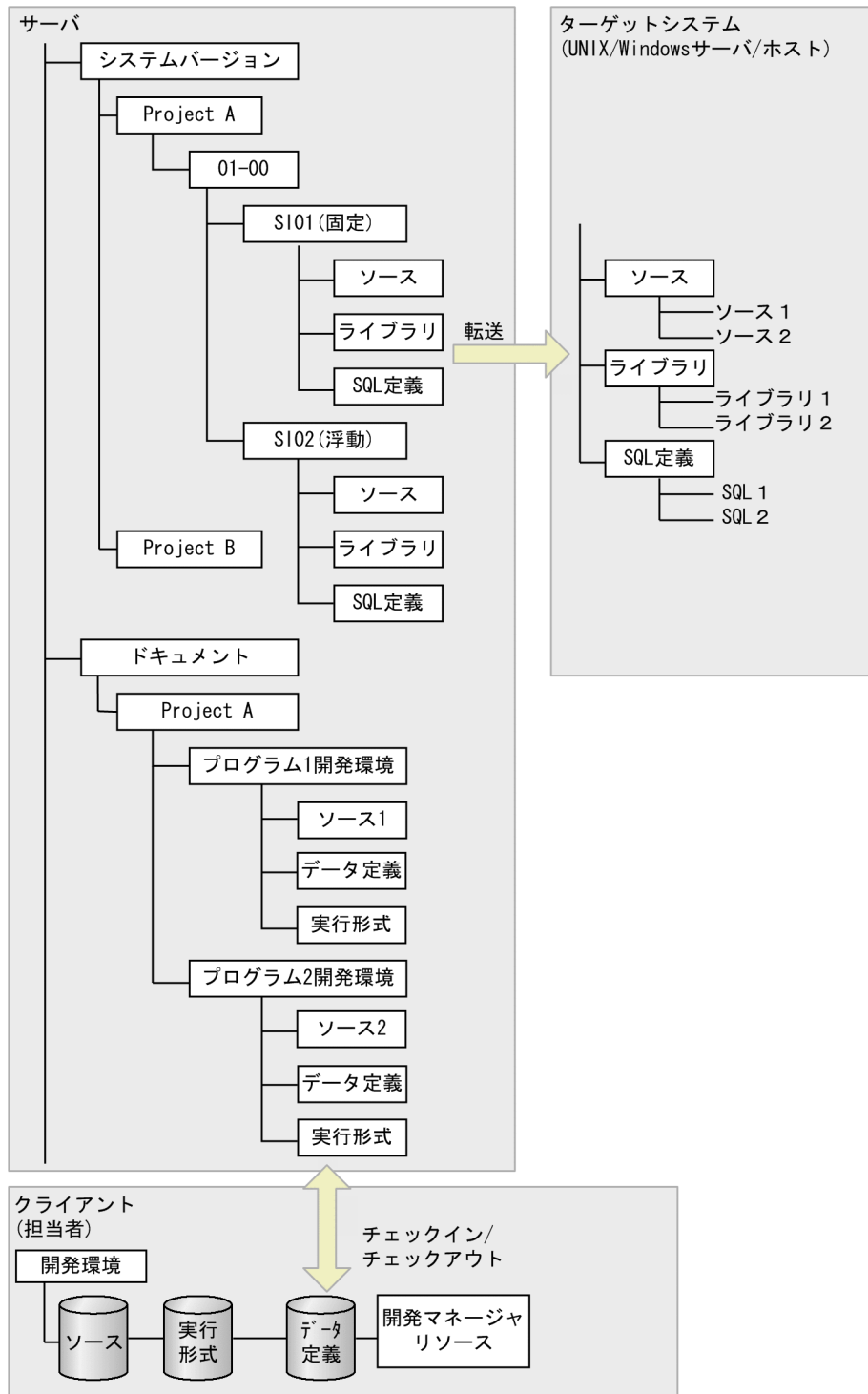
10.1.1 転送を考慮したドキュメントフォルダ構成

統合テスト時には、Windows からターゲットシステムへ、テストの対象になるドキュメントを転送します。統合テストをするためのシステム統合では、個々に作成されたプログラムを統合し、一つのプログラムとして組み立てるので、それぞれのプログラムの同期や修正履歴を採る必要があります。

(1) システムバージョンの利用

SEWB+/REPOSITORY のシステムバージョンを利用して、次の図のようなドキュメントフォルダ構成で運用し、ターゲットシステムへ生成物を転送してください。システムバージョンについては、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

図 10-1 転送を考慮したドキュメントフォルダ構成



(2) SI 支援ユティリティの利用

(a) SI 支援ユティリティとは

SI (System Integration) 支援ユティリティは、システムバージョンを利用して、サーバ上で異なるシステムバージョンに関連づけられたドキュメントを比較し、抽出します。統合テスト用にターゲットシステムへファイルを転送する場合、SI 支援ユティリティを利用すると、特定のシステムバージョンに関連づけ

られたドキュメントのファイル内容だけをローカルフォルダに抽出できるので、転送するファイルを容易に準備できます。

なお、SI 支援ユティリティはリポジトリ管理者だけが実行できます。SI 支援については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

(b) SI 支援ユティリティの実行条件

SI 支援ユティリティを利用するための実行条件を次に示します。

- 実行場所
SI 支援ユティリティは、サーバで実行します。クライアントでは、実行できません。
- 対象となる資源
システムバージョンの設定されているドキュメントだけを対象としています。システムバージョン単位に対象、非対象を指摘できます。したがって、システムバージョンが設定されていない場合、SI 支援ユティリティは使用できません。
- 権限
SI 支援ユティリティを実行できるのは、システム管理者だけです。
- アクセス権
SI 支援ユティリティは、システム管理者が実行するため、資源に設定されているアクセス権に関係なく実行できます。
- 排他処理
SI 支援ユティリティは、排他処理を行っていません。そのため、SI 支援ユティリティの実行中に、ほかの処理を並行して実行できます。

10.2 ターゲットシステムでのテストと修正

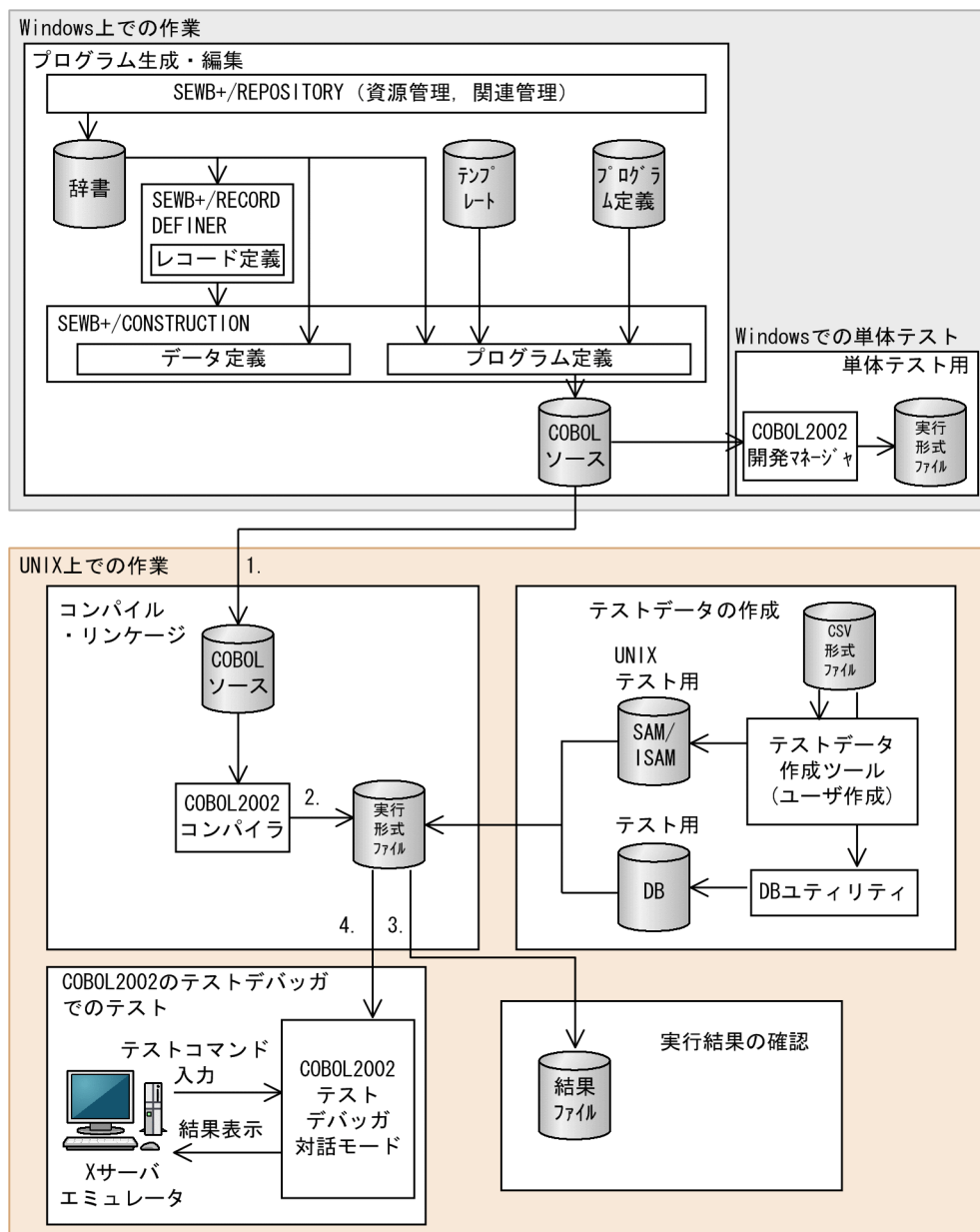
ターゲットシステム上でのテスト手順とソースファイルの修正手順について説明します。

10.2.1 ターゲットシステムが UNIX の場合

(1) テスト手順

ターゲットシステムが UNIX の場合の統合テストの例を次の図に示します。なお、開発言語は COBOL2002 を想定しています。

図 10-2 ターゲットシステムが UNIX の場合の統合テストの例



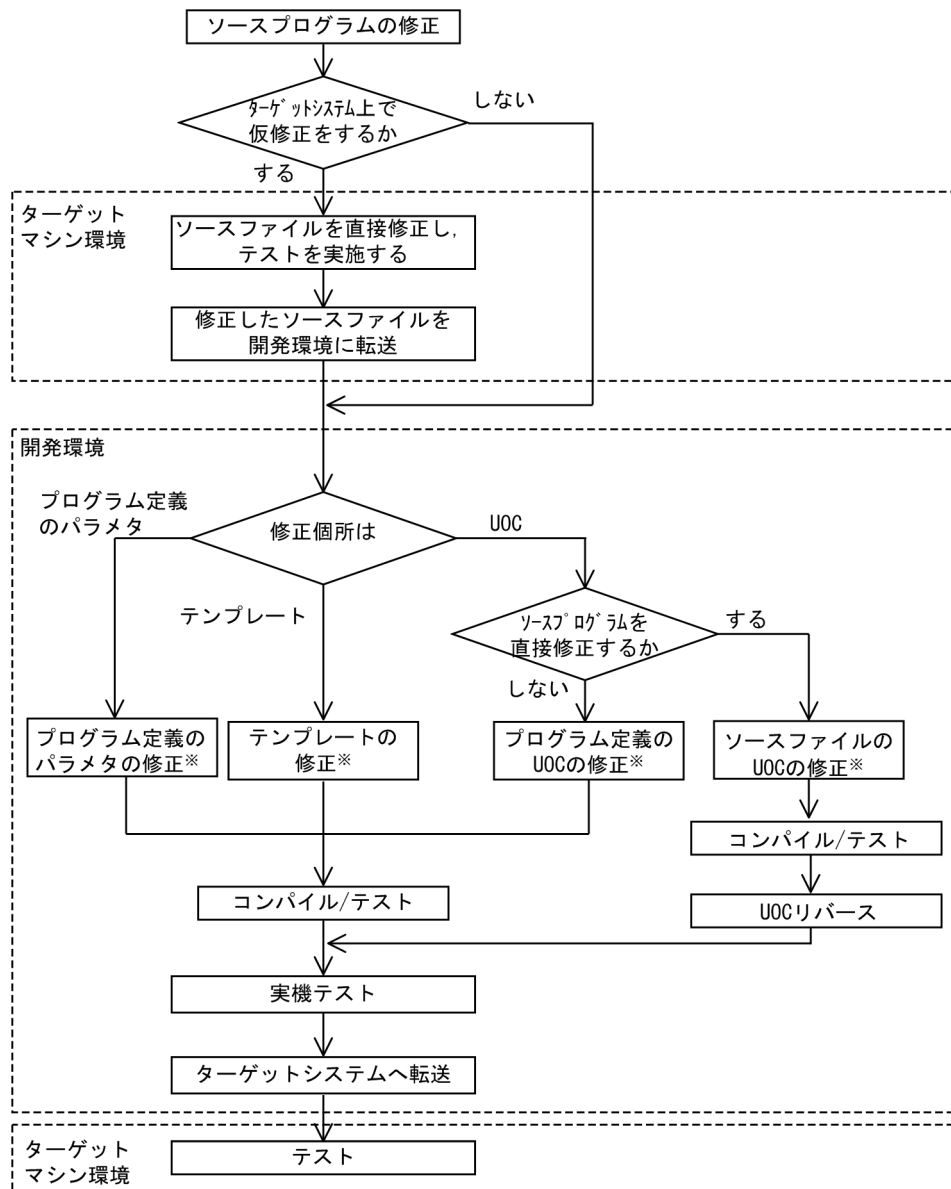
1. Windows から UNIX へ生成物を転送する。

2. ターゲットシステム上でコンパイルし，実行形式ファイルを作成する。
3. 統合テストとして，プログラムの結果を確認する。
4. 結果が不正な場合やプログラムが異常終了したときは，デバッグオプションを指定し，対話モードでプログラムの不良原因を調べる。

(2) ソースファイルの修正手順

ターゲットシステム上でテスト時に不良を発見したり，仕様変更が発生したりした場合，ソースファイルを修正する必要があります。ソースファイルを修正する場合は，次の図に示す手順に従ってください。

図 10-3 ソースファイルの修正手順



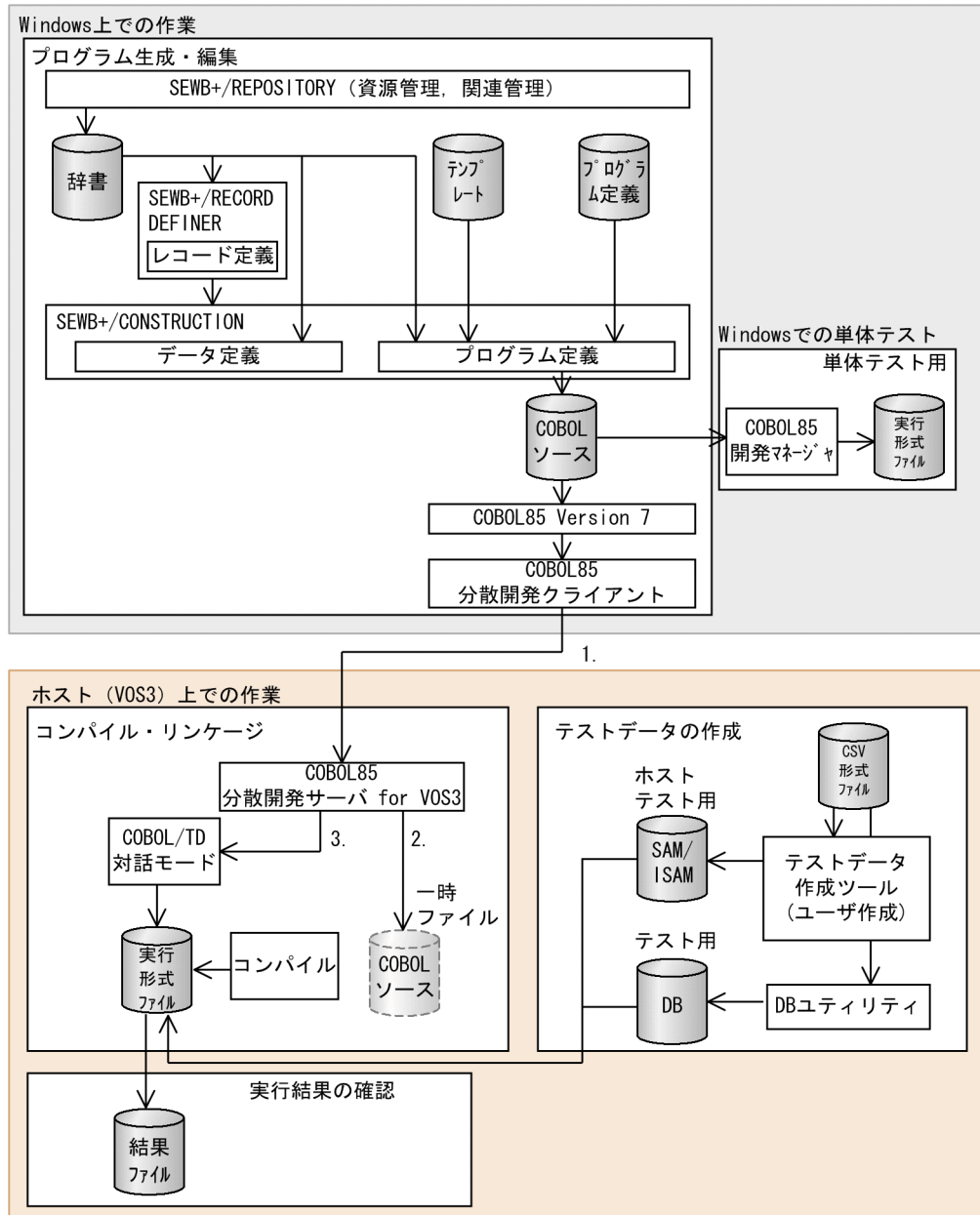
注

これらの修正の手順については，「9.1.3 プログラムの修正」を参照してください。

10.2.2 ターゲットシステムがホストの場合

(1) テスト手順

ターゲットシステムがホストの場合の統合テストの例を次に示します。なお、開発言語は COBOL85 Version 7 を想定しています。



1. Windows 上の COBOL85 開発マネージャを利用し、ホストへ生成物を転送する。
2. ホスト (VOS3) 上の COBOL85 開発マネージャからコンパイル・リンケージし、実行形式ファイルを作成する。
3. ホスト (VOS3) 上の COBOL85 開発マネージャから対話モードでデバッグをする。

(2) ソースファイルの修正手順

ターゲットシステム上でテスト時に不良を発見したり，仕様変更が発生したりした場合，ソースファイルを修正する必要があります。ソースファイルを修正する場合の手順を次に示します。

1. SEWB+/REPOSITORY からプログラム定義をチェックアウトし，修正する。
2. プログラム定義の修正後，ソースファイルを再生成する。

注

再生成したソースファイルはホスト（VOS3）上には作成されないので，ソースファイルを直接修正することはできません。

11 資源の管理

SEWB+ では、システム開発に使用する資源を一括管理するリポジトリとして SEWB+/REPOSITORY を、リポジトリで管理されている資源を閲覧したり操作したりする機能として SEWB+/REPOSITORY-BROWSER を使用します。この章では、SEWB+/REPOSITORY での資源管理について説明します。

11.1 リポジトリの機能

11.2 リポジトリの準備

11.3 リポジトリでの命名時の注意事項

11.1 リポジトリの機能

リポジトリで管理する資源には、ドキュメントと辞書があります。それぞれはフォルダで管理されています。フォルダとは、ディレクトリに相当する入れ物です。

11.1.1 フォルダでのドキュメント管理と辞書管理

リポジトリではフォルダで資源を分類して管理します。ドキュメントを格納するフォルダをドキュメントフォルダといい、辞書を格納するフォルダを辞書フォルダといいます。

(1) ドキュメントフォルダ

ドキュメントフォルダの場合、最上位にはルートドキュメントフォルダがあります。ルートドキュメントフォルダは、システム開発のプロジェクト単位にシステム管理者が作成します。ルートドキュメントには、リポジトリブラウザを使ってドキュメントフォルダやドキュメントを作成して登録します。このようにすると、ドキュメントを階層構造で管理できます。

ドキュメントを作成するには、次の二つの方法があります。

- ドキュメントを新規作成する
- クライアントのファイルシステムにあるファイル（ローカルファイル）をドキュメントとしてリポジトリに登録する

なお、これらの作業の前に、システム管理者が利用環境を設定しておく必要があります。利用環境の設定については、「11.2 リポジトリの準備」またはマニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

(2) 辞書フォルダ

辞書フォルダは、プロジェクト単位にシステム管理者が作成します。辞書フォルダには、データ項目辞書および業務ルール辞書を格納します。なお、辞書フォルダは階層構造で管理できません。

また、辞書は常に最新の状態で更新されるので、バージョンや更新履歴で管理する必要はありません。

11.1.2 ドキュメントの管理

(1) システムバージョンでの管理

リポジトリで管理する同じプロジェクト内のドキュメント同士でも、システム開発工程のある時点では、それぞれの更新回数（版の数）が異なります。このため、システム開発に関係する複数のドキュメントに対して、ある時点でのドキュメント間の整合性が取れているように管理することが重要です。SEWB+/REPOSITORY では、個々のドキュメントの版に対して、特定のバージョン名を設定できます。この特定のバージョン名をシステムバージョンといい、設定された時点の版を示す「固定」と、設定した時点以降に更新された最新の版を示す「浮動」に区別されます。システムバージョンを設定したり解除したりするための権限は、システム管理者や資源の所有者が設定しておきます。権限の設定については、「11.2 リポジトリの準備」を参照してください。

(2) 更新履歴での管理

それぞれのドキュメントは、リポジトリ内に更新履歴を保持します。SEWB+/REPOSITORY では、更新ごとのドキュメントを管理します。

11.1.3 ドキュメントの更新

リポジトリで管理しているドキュメントを更新するには、更新専用のチェックアウトでリポジトリのドキュメントの複製をクライアントのファイルシステム上に取り出します。チェックアウト時には、任意のバージョンを指定できます。ただし、更新専用チェックアウトされているドキュメントに対しては、新たに更新専用チェックアウトできません。

なお、更新専用でチェックアウトしたドキュメントは、更新後、リポジトリにチェックインで返却します。このとき、チェックインするドキュメントとリポジトリ内のドキュメントの更新日が比較され、リポジトリ内のドキュメントの更新日がチェックインするものと同じかまたは古い場合は、ドキュメントを再登録するかしないかの選択ができます。

注

SEWB+/CONSTRUCTION で使用するドキュメントを更新する場合、ドキュメント間に付けられている関連を保持するために、適切な手順でチェックインおよびチェックアウトする必要があります。修正時の手順については、「8.3.2 修正時の手順」を参照してください。

11.1.4 ドキュメントの参照

リポジトリで管理しているドキュメントを参照するには、参照専用のチェックアウトを利用するか、ネットワーク機能を利用します。

参照専用チェックアウト

参照専用チェックアウトをして、ドキュメントの複製をクライアントのファイルシステムに取り出します。参照専用チェックアウトをしたドキュメントは、リポジトリにチェックインできません。なお、一つのドキュメントに対して、同時に複数の参照専用チェックアウトができます。

ネットワーク機能

リポジトリのルートドキュメントフォルダは OS 上で共有宣言されているため、チェックアウトしなくてもエクスプローラやツールから検索、参照できます。ただし、参照できるドキュメントは、最新のバージョンだけです。また、Windows Server のネットワーク機能を使用するため、Windows Server のアカウントが必要です。詳細は、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

11.1.5 関連の設定

リポジトリで管理する資源間の依存関係を表すために、関連を付けることができます。関連には、SEWB+ が自動的に関連づけるものと、ユーザが任意に付けるものとがあります。ユーザが任意に付ける関連の種別をユーザ関連といいます。

資源に関連を付けておくと、関連ブラウザを使用して関連づけられた資源をブラウジングできるので、変更波及の調査などに役立ちます。

表 11-1 資源と関連の関係

関連を付ける機能または製品	関連する資源	関連の設定時	関連種別
SEWB+/CONSTRUCTION	プログラム定義とテンプレート	プログラム定義の登録またはチェックイン時	参照
	プログラム定義と部品		
	プログラム定義とデータ定義		
	プログラム定義とマップ定義ファイル		

関連を付ける機能または製品	関連する資源	関連の設定時	関連種別
	プログラム定義と XML 文書ファイル		
	プログラム定義と業務ルール		辞書参照
	プログラム定義と生成ソース		生成
	データ定義と辞書のレコード定義（最上位結合項目）	データ定義の登録またはチェックイン時	辞書参照
	データ定義と SEWB+/RECORD DEFINER のレコード定義		参照
	データ定義と生成ソース		生成
SEWB+/RECORD DEFINER	レコード定義とデータ項目	レコード定義の登録またはチェックイン時	辞書参照
	レコード定義と生成ソース		生成
SEWB+/REPOSITORY	業務ルールとデータ項目	業務ルール作成時	辞書参照
	単一項目と結合項目	結合項目作成時	結合
	単一項目と継承項目	継承関係の下位のデータ項目作成時	継承
EUR	帳票仕様書とデータ項目	ドキュメントの登録またはチェックイン時	辞書参照

注 1

ドキュメント間の関連が設定されるのは、参照するドキュメントがネットワークドライブで参照されているファイル、または更新目的でチェックアウトされているファイルの場合です。参照目的でチェックアウトされたファイル、または SEWB+/REPOSITORY に登録されていないファイルには、関連は設定されません。

注 2

SEWB+/CONSTRUCTION、SEWB+/RECORD DEFINER で複数の資源に関連を設定したい場合は、SEWB+/REPOSITORY に一括してチェックインするようにしてください。一括でチェックインできないときは、プログラム定義ファイルを最初にチェックインしてください。

11.1.6 アクセス権の設定

設定しておくアクセス権を次に示します。

- リポジトリのアクセス権
- ネットワーク機能利用のアクセス権

(1) リポジトリのアクセス権

リポジトリの資源を作成したり、更新したりする権限を設定しておきます。アクセス権は、システム管理者または資源の所有者が設定します。

(a) アクセス権を設定する資源

アクセス権はリポジトリブラウザを使用して設定します。ただし、ルートドキュメントフォルダと辞書フォルダのアクセス権はリポジトリの環境構築ユーティリティで設定します。次に資源とアクセス権の関係を示します。なお、アクセス権を指定する場合、「変更」のように幾つかのアクセス権を組み合わせて指定するものがあります。「変更」の組み合わせは資源ごとに異なります。「フルコントロール」はその資源に与えられているすべてのアクセス権を含んでいます。

ルートドキュメントフォルダ (環境構築ユティリティ設定)

アクセス権		内容
変更	作成	下位オブジェクト (ドキュメントフォルダやドキュメント) の作成
	読み取り	<ul style="list-style-type: none"> ルートドキュメントフォルダの表示 下位オブジェクト (ドキュメントフォルダやドキュメント), プロパティなどの設定情報の参照
	書き込み	プロパティなどの設定情報の更新
削除		フォルダの削除

ドキュメントフォルダ

アクセス権		内容
変更	作成	下位オブジェクト (ドキュメントフォルダやドキュメント) の作成
	読み取り	<ul style="list-style-type: none"> ドキュメントフォルダの表示 プロパティなどの設定情報の参照
	書き込み	プロパティなどの設定情報の更新
削除		フォルダの削除

システムバージョンフォルダ

アクセス権		内容
変更	作成	下位オブジェクト (システムバージョンフォルダやシステムバージョン) の作成
	読み取り	<ul style="list-style-type: none"> システムバージョンフォルダの表示 下位オブジェクト (システムバージョンフォルダやシステムバージョン), プロパティなどの設定情報の参照
	書き込み	プロパティなどの設定情報の更新
削除		システムバージョンの削除

システムバージョン

アクセス権		内容
変更	設定	システムバージョンのドキュメントへの設定および解除
	読み取り	<ul style="list-style-type: none"> システムバージョンの表示 プロパティなどの設定情報の参照
	書き込み	プロパティなどの設定情報の更新
削除		システムバージョンの削除

ドキュメント

アクセス権		内容
変更	読み取り	<ul style="list-style-type: none"> ドキュメントの表示 ドキュメントの内容, プロパティなどの設定情報の参照
	書き込み	ドキュメントの内容, プロパティなどの設定情報の更新
削除		ドキュメントの削除

辞書フォルダ（環境構築ユティリティ設定）

アクセス権		内容
作成		下位オブジェクト（データ項目や業務ルール）の作成
変更	読み取り	下位オブジェクト（データ項目や業務ルール）、プロパティなどの設定情報の参照
	書き込み	プロパティなどの設定情報の更新
削除		辞書フォルダの削除

データ項目

アクセス権		内容
変更	読み取り	<ul style="list-style-type: none"> データ項目の表示 データ項目の定義内容、プロパティなどの設定情報の参照 下位データ項目の作成
	書き込み	データ項目の定義内容、プロパティなどの設定情報の更新
削除		データ項目の削除

業務ルール

アクセス権		内容
変更	読み取り	<ul style="list-style-type: none"> 業務ルールの表示 業務ルールの定義内容、プロパティなどの設定情報の参照
	書き込み	業務ルールの定義内容、プロパティなどの設定情報の更新
削除		業務ルールの削除

(b) 使用者の種類

使用者は「システム管理者」、「所有者」、「グループ」または「全ユーザ」に分けられ、それぞれにアクセス権を設定します。アクセス権には「読み取り」、「書き込み」、「作成」、「設定」および「削除」があります。ただし「設定」はシステムバージョンだけに有効な権限です。

(2) ネットワーク機能利用のアクセス権

リポジトリに管理されている資源は、Windows Server のネットワーク機能でも参照できます。ネットワーク機能を利用するには、Windows Server のアカウントが必要です。また、リポジトリを利用するには、SEWB+/REPOSITORY のアカウントが必要です。

なお、リポジトリに管理されている資源を利用する場合には、SEWB+/REPOSITORY および Windows のアカウント名（ユーザ名、グループ名）を同一にする運用をお勧めします。

異なるアカウント名を使用すると、ドキュメントの参照ができないなどの現象が起きることがあります。

アカウント名の設定手順については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

11.2 リポジトリの準備

リポジトリで資源を管理するために、あらかじめ利用環境を設定しておきます。ここでは、あらかじめどのような作業が必要かを説明します。リポジトリの環境設定については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

11.2.1 サーバ側およびクライアント側の環境設定

サーバ側およびクライアント側の環境設定が必要です。

(1) サーバ側での環境設定

サーバ側の環境設定には「SEWB+/REPOSITORY の前提プログラムの環境設定」、「環境の初期化」、「SEWB+/REPOSITORY のユーザ登録」および「運用環境の構築」があります。

(a) SEWB+/REPOSITORY の前提プログラムの環境設定

Windows Server での環境設定、および DB での環境設定をします。

Windows Server での環境設定

SEWB+/REPOSITORY では、ネットワーク環境に TCP/IP を使用しているため、SERVICES ファイルに SEWB+/REPOSITORY 用の設定をします。この設定は必須です。

DB での環境設定

リポジトリを格納する DB (Object Server) の環境は必要に応じてシステム管理者が設定してください。SEWB+/REPOSITORY を使用するためには、初期設定パラメタファイルおよびシステム共通定義ファイルを作成してください。これらのファイル作成は必須です。

初期設定パラメタファイルは DB の初期設定情報を記述するファイルです。

システム共通定義ファイルは、DB のシステムに共通する定義情報を記述するファイルです。

SEWB+/REPOSITORY の環境を設定する場合のそれぞれのファイルの指定例は、サンプルファイルで提供されています。サンプルファイルをコピーして利用してください。

(b) 環境の初期化

DB およびリポジトリを初期化します。

DB の初期化

Object Server を初期化します。この作業は必須です。DB の初期化時には必ずノード ID を指定してください。ただし、SEWB+/REPOSITORY では、エクスポート・インポート機能で複数のサーバ間で資源をやり取りします。指定するノード ID が、複数のサーバの DB 間で重複しないように注意してください。また、DB を格納するディレクトリのファイルシステムには、必ず NTFS を使用してください。FAT は使用できません。

リポジトリの初期化

DB 名称を設定したり、ドキュメントオプションを設定したりします。DB 名称の設定は必須です。なお、DB 名称の規則については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

(c) ドキュメント種別の登録

SEWB+ 以外のツールで作成するドキュメントをリポジトリで管理する場合は、あらかじめ環境構築ユーティリティでドキュメント種別を登録しておく必要があります。また、ドキュメント種別の登録方法には、

SEWB+ ツールからの自動登録，または環境情報のインポートによる登録もあります。複数サーバで運用する場合は，環境情報のインポートによる登録をお勧めします。

注

環境構築ユーティリティでは，テンプレートファイルのパスも設定します。テンプレートファイルパスを省略すると，リポジトリブラウザからドキュメントの新規作成ができなくなります。ただし，省略時でも，ドキュメント登録はできます。

(d) SEWB+/REPOSITORY のユーザ登録

ユーザ登録ファイルおよびグループ登録ファイルを設定します。ユーザ登録ファイルの設定は必須です。Windows Server のネットワーク機能を利用してドキュメント参照をする場合は，SEWB+/REPOSITORY と Windows Server に設定するアカウント名（ユーザ名，グループ名）を同一にしてください。

ユーザ登録ファイルの設定

SEWB+/REPOSITORY のユーザを登録するファイルです。

ポイント

SEWB+/REPOSITORY へ同時にログインできるのは，system を含めて 230 ユーザまでです。ただし，OS のリソースおよび設定値によって異なりますが，同時にログインできるユーザは，40 ユーザまでにすることを推奨します。

グループ登録ファイルの設定

SEWB+/REPOSITORY のユーザのグループを登録するファイルです。標準のグループとして，SEWB+/REPOSITORY の管理者グループである「system」とすべてのユーザが所属するグループである「everyone」があります。そのほかのグループは，必要に応じて追加してください。

(e) 運用環境の構築

運用環境は，環境構築ユーティリティで設定します。環境構築ユーティリティで設定できる項目を次の表に示します。

表 11-2 環境構築ユーティリティの設定項目

環境構築ユーティリティの設定項目	設定の要否
システム管理者用のパスワード設定	
ルートドキュメントフォルダの作成	
辞書フォルダの作成	
同時にログインできるユーザ数の設定	
操作ログの設定	
ドキュメントの設定	
バックアップ用コマンドファイルの作成	
SEWB+ クライアントプログラムをターミナルサービス クライアントから使用するかどうかの設定	
データ項目と業務ルールの設定	
辞書ユーティリティタイプマッピングの設定	
ドキュメント種別の設定	

(凡例)

- : 必ず設定する
- : 設定することを推奨する
- : 必要に応じて設定する

注

Windows Server 2008 R2 のリモートデスクトップサービスを示します。

(2) クライアント側での環境設定

クライアント側の環境設定には「SEWB+/REPOSITORY-BROWSER の前提プログラムの環境設定」と「運用開始時の設定」があります。

(a) SEWB+/REPOSITORY-BROWSER の前提プログラムの環境設定

リポジトリブラウザの稼働に必要な、Windows での環境として、通信設定ファイル (HOSTS , SERVICES) を設定します。この作業は必須です。

(b) 運用開始時の設定

リポジトリブラウザで一般ユーザのパスワードを設定します。設定は任意ですが、機密保護のために設定することをお勧めします。

11.3 リポジトリでの命名時の注意事項

リポジトリで管理する資源に名前を付けるときは、ルートドキュメントフォルダ名に「+」を使用しないでください。

12 ドキュメントの運用

SEWB+/REPOSITORY では、ドキュメントを効率良く運用できるように環境を設定できます。この章では、ドキュメントを効率良く運用するためのポイントについて説明します。

12.1 環境設定

12.2 ドキュメントフォルダの作成と運用

12.3 セキュリティ管理

12.4 ドキュメント作成と修正の手順

12.1 環境設定

ドキュメントを効率良く作成したり管理したりするために、環境を設定します。リポジトリの環境構築ユティリティで、システム管理者が環境を設定します。

(1) ルートドキュメントフォルダの作成

ドキュメントフォルダを格納するルートドキュメントフォルダを作成します。

(2) ドキュメントの設定

ドキュメント、ドキュメントフォルダ、システムバージョンおよびシステムバージョンフォルダに設定されるアクセス権の初期値を設定できます。アクセス権を継承するように設定して、資源のアクセス権を一括で設定したりできます。

(3) ドキュメント種別の登録

リポジトリで管理したいドキュメントの種別を登録します。効率良くドキュメント作成できるよう、あらかじめ登録しておくことをお勧めします。

12.2 ドキュメントフォルダの作成と運用

ドキュメントを格納するフォルダの作成方法と運用方法について説明します。

(1) ルートドキュメントフォルダの作成と運用

ルートドキュメントフォルダは OS 上で共有宣言されているフォルダです。

ルートドキュメントフォルダは運用上、次に示す作業の単位になります。作成時には、これらを考慮してください。

- インポート・エクスポートの単位
- バックアップ・リストア（環境回復ユティリティ）の単位
- ネットワークドライブの共有フォルダ

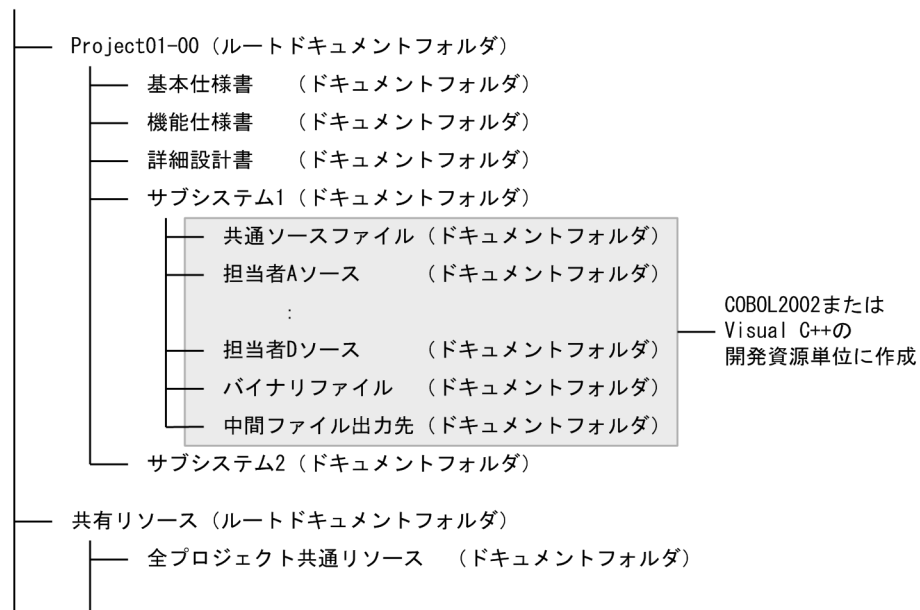
(2) ドキュメントフォルダおよびサブドキュメントフォルダの作成

ローカルの作業領域で使用する資源を洗い出し、どういう単位で管理するかを検討してからフォルダ構成を決めます。なお、資源の分類別にドキュメントフォルダを作成し、管理する個々のドキュメントをサブドキュメントフォルダとして格納すると、一括してチェックイン・チェックアウトができるので便利です。

(3) ドキュメントフォルダの構成例

ドキュメントフォルダの構成例を次の図に示します。

図 12-1 ドキュメントフォルダの構成例



12.3 セキュリティ管理

ドキュメント情報を保護するためのアクセス権と排他制御について説明します。

(1) アクセス権の設定

複数のグループで共同作業をする場合など、リポジトリで管理するドキュメントにアクセス権を設定して、資源を共有したり、機密を保護する必要があります。次にアクセス権の設定の組み合わせと、設定した結果がどのようなになるかを示します。

(a) デフォルトの設定値を利用する

SEWB+/REPOSITORY のアクセス権のデフォルトを利用すると、ドキュメントを次のように管理できます。

デフォルトのアクセス権

資源	所有者	グループ名	アクセス権		
			所有者	グループ	全ユーザ
ドキュメントフォルダ	環境構築ユティリティの「ドキュメント」で設定した内容	ルートドキュメントで指定した内容	フルコントロール	読み取り	読み取り
ドキュメント					

デフォルトのアクセス権を設定した場合

ユーザ例	アクセス権			
	作成	変更	削除	参照
A グループのドキュメント作成者				
A グループのドキュメント作成者以外のメンバ	x	x	x	
A グループ以外	x	x	x	

(凡例)

- : 権利がある
- x : 権利がない

(b) カスタマイズ例

プロジェクトでの運用に合わせて、デフォルトで設定されている値をカスタマイズしてください。

ただし、ドキュメントフォルダやドキュメントのアクセス権を設定する場合、個々に設定をすると効率が悪くなります。ドキュメントフォルダとドキュメントのアクセス権を同じにしたい場合は、ドキュメントフォルダのアクセス権がそのままドキュメントへ継承されるように、あらかじめリポジトリの環境構築ユティリティで設定しておきます（ドキュメントの初期属性の設定）。このようにすると、ドキュメントフォルダのアクセス権を設定するだけで、格納されているすべてのドキュメントのアクセス権を設定できるようになります。

ドキュメントの初期属性の設定方法については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

次に、カスタマイズ例を示します。

アクセス権の設定

資源	所有者	グループ名	所有者 アクセス	グループ	全ユーザ
ドキュメントフォルダ	ドキュメント作成者	Aグループ	フルコントロール	フルコントロール	読み取り
ドキュメント					

前述のようにアクセス権を設定した場合

ユーザ例	アクセス権			
	作成	変更	削除	参照
Aグループのドキュメント作成者				
Aグループのドキュメント作成者以外のメンバ				
Aグループ以外	x	x	x	

(凡例)

：権利がある

x：権利がない

(2) 排他制御

複数のユーザに共用されるドキュメントは、同時に更新されるおそれがあります。SEWB+/REPOSITORYでは、リポジトリに格納されたドキュメントの同時更新を防ぐために、OSの排他制御機能のほかに、更新専用チェックイン・チェックアウト機能で排他制御をしています。

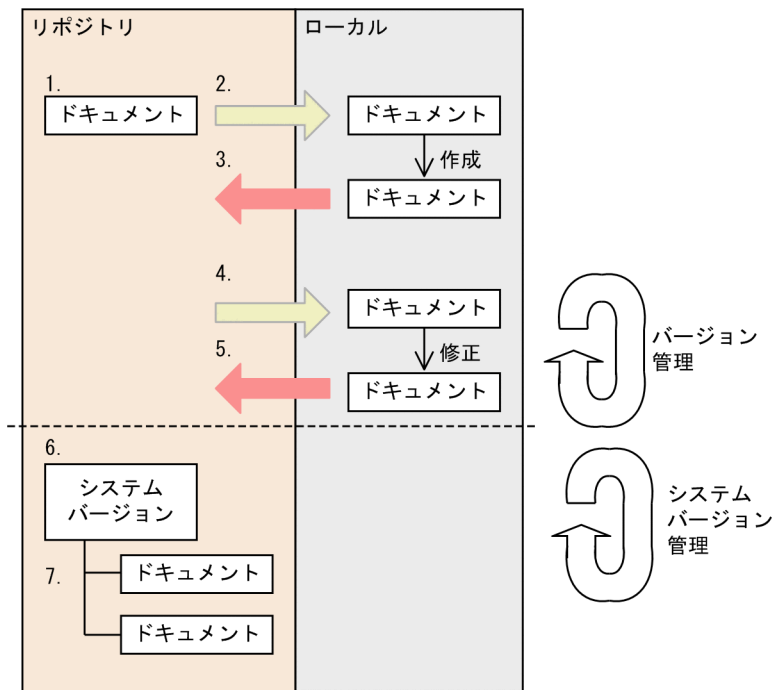
12.4 ドキュメント作成と修正の手順



リポジトリで管理するドキュメントを修正する場合は、必ず更新専用チェックアウトします。チェックイン・チェックアウトされたドキュメントは、SEWB+/REPOSITORY が付けるバージョン（版）で管理されます。

また、テスト終了時など、作業の大きな区切りにはシステムバージョンを作成して、システムバージョン管理をします。

ドキュメント作成と修正の手順を次の図に示します。

図 12-2 ドキュメント作成と修正の手順



(凡例)  : チェックアウト
 : チェックイン

- リポジトリにドキュメント種別を登録する。
- SEWB+/REPOSITORY-BROWSER からドキュメントを新規作成する。
新規作成するとドキュメントがリポジトリに登録され、そのあと、チェックアウトされます。これが 1 回目のチェックアウトです。
- ドキュメントを作成し、終了したらチェックインする。
- 修正など、必要に応じて更新専用のチェックアウトをする。
- 更新専用チェックアウトしたものは、必ずチェックインする。
4, 5. の作業のように、作業の節目ではなく作成作業の途中で発生するようなチェックアウト・チェックインは、バージョンで管理されます。
- システムバージョンを作成する。
テスト終了時などの作業の節目でドキュメントの最新バージョンや固有のバージョンを管理する必要がある

出てきたら，システムバージョンを作成し，管理したいドキュメントに対して設定します。

7. システムバージョンを，管理したいドキュメントに設定する。

13 辞書の運用

SEWB+/REPOSITORY では、プロジェクトやシステム開発環境に応じて、辞書を運用しやすくするために環境を設定できます。また、辞書が格納するデータ項目の特徴によって辞書を分類して管理できます。この章では、辞書を効率良く運用するためのポイントについて説明します。

13.1 環境設定

13.2 辞書フォルダの分割作成と構成

13.3 セキュリティ管理

13.1 環境設定

システム管理者は、辞書を効率良く運用するため、プロジェクトや開発体制に応じて環境を設定します。環境を設定するときは、リポジトリの環境構築ユティリティを使用します。

13.1.1 必要な環境設定

用途によって必要になる環境設定について、代表的なものを示します。

(1) 言語区分

アプリケーション開発に使用するプログラミング言語の名称を設定します。

(2) ERwin/ERX タイプのマッピング

ERwin/ERX や ODBC から SEWB+/REPOSITORY にデータ項目を取り込むときに適用されるタイプマッピングを、必要に応じてカスタマイズできます。次にカスタマイズの例を示します。

(カスタマイズが必要な場合の例)

カスタマイズ前

DB に ORACLE を使用している場合、ORACLE の標準のタイプマッピング (変換規則 A) で CSV 形式ファイルからデータ項目を取り込むと、次のようにマッピングされます。

ERwin/ERX のタイプ	SEWB+/REPOSITORY でのマッピング		
	分類	言語別タイプ (COBOL)	けた数
NUMBER(p [, 0])	なし	X(英数字項目)	p+2

これをソース生成すると、「データ名 PIC X(p+2)」となります。

カスタマイズ内容

取り込んだデータ項目を整数として扱いたい場合は「NUMBER(p [, 0])」のマッピングを次のようにカスタマイズする必要があります。

ERwin/ERX のタイプ	SEWB+/REPOSITORY でのマッピング		
	分類	言語別タイプ (COBOL)	けた数
NUMBER(p [, 0])	整数	P(符号付き内部 10 進項目)	p

このようにマッピングされたデータ項目をソース生成すると「データ名 PIC S9(p) USAGE PACKED - DECIMAL」となります。

(3) レコード生成キーワード

レコード生成キーワードには、データ項目の定義情報をどのようにソースコードに生成するかを指定します。例えば、レコード生成キーワードのデフォルトの指定は、COBOL2002 を前提にしています。そのため、COBOL2002 以外の COBOL を使用する場合は、タイプごとに生成するソースの文法に相違がないか確認し、必要であればレコード生成キーワードを変更します。

注

この指定は、SEWB+/CONSTRUCTION でのすべてのプログラム生成およびレコードソース生成に有効になります。編集に際しては、あらかじめ十分な検討が必要です。

13.2 辞書フォルダの分割作成と構成

SEWB+ でのシステム開発では、全体で一つの辞書フォルダを使用する運用もできますが、開発形態や体制などを考慮して、複数の辞書フォルダに分割して使用する運用もできます。

13.2.1 辞書分割時の注意事項

辞書を分割するときは、次に示す点に注意してください。

- 異なる辞書フォルダ間の継承はできない。
- 一つの辞書に格納できるデータ項目は 3 万個。
- 辞書フォルダは階層化できない。

13.2.2 データ項目の分類方法

辞書を分割して運用する場合のデータ項目の分類方法を、次に示します。

- 抽出元（データのソース）での分類
- ドメインでの分類
- アクセスする人、グループなどでの分類
- 結合関係での分類

(1) 抽出元（データのソース）での分類

何から抽出されたデータ項目かによって分類します。主に DB 設計で抽出されたデータ項目とそれ以外の項目に分けられます。

(a) 抽出元での分類の利点

抽出元でデータ項目进行分类すると、次のような利点があります。

- データ項目の抽出者ごとの管理対象を明確にできる。
- ERwin/ERX からデータ項目を SEWB+/REPOSITORY に取り込む際に、取り込み先を独立させられるので、ERwin/ERX と辞書との整合性の確認などが効率良くできる。

(b) 分類例

分類	辞書に格納するデータ項目
RDB 項目	RDB のテーブル（最上位結合項目）およびカラム（単項目）
RDB 以外の項目	<ul style="list-style-type: none"> • ファイルや電文などで使用する項目（結合項目，単項目） • ファイルや電文のレコード（最上位結合項目） RDB 以外の項目は、ファイルや電文などさらに細かく分類できる。

(2) ドメインでの分類

ドメインになるデータ項目およびドメインを継承するデータ項目を一つの辞書フォルダにまとめることで、ドメインを利用しないデータ項目と区別します。

(a) ドメインでの分類の利点

継承は、異なる辞書フォルダ間ではできません。そのため、継承関係を結びたいデータ項目が、本来の分類ではない分類の辞書フォルダに格納されてしまうことがあります。これを避けるため、ほかの分類とは独立したドメインだけの辞書フォルダを作成します。

(b) 分類例

分類	辞書に格納するデータ項目
ドメイン項目	ドメインになるデータ項目およびドメインを継承するデータ項目
非ドメイン項目	ドメインにならないデータ項目で、ドメインを利用しないデータ項目

(3) アクセスする人、グループなどでの分類

データ項目を作成する人、データ項目を参照する人およびそれらのグループによって分類します。

(a) アクセスする人、グループなどでの分類の利点

アクセスする人、グループなどでデータ項目进行分类すると、次のような利点があります。

- 作成者および作成グループの管理対象を明確にできる。
- 参照者および参照グループが効率良く検索できる。
- アクセス権管理者の単位に分類できる。

(b) 分類例

分類	辞書に格納するデータ項目
共通項目	すべての業務で共通に使用するデータ項目
業務別項目	特定の業務だけで使用するデータ項目

(4) 結合関係での分類

最上位結合項目とその構成項目で分類します。

(a) 結合関係での分類の利点

結合関係でデータ項目进行分类すると、次のような利点があります。

- データ分析者（単項目作成者）とレコード作成者（最上位結合項目作成者）の操作対象を明確にできる。
- 最上位結合項目を作成するときに、対象になる構成項目の絞り込みができる。

(b) 分類例

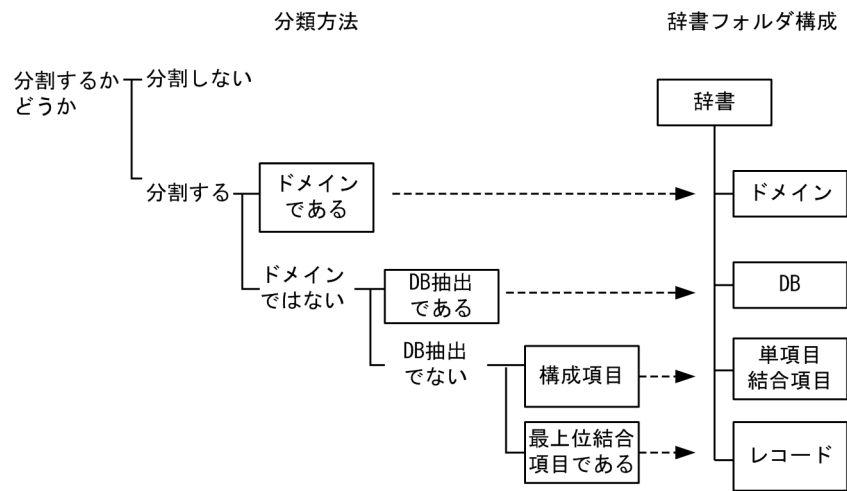
分類	辞書に格納するデータ項目
単項目、結合項目	<ul style="list-style-type: none"> • 単項目 • 最上位結合項目以外の結合項目 データ分析者が作成し、レコード作成者が参照する。
レコード	最上位結合項目 レコード作成者が作成する。

13.2.3 分類方法と辞書フォルダ構成

辞書フォルダを分割する場合、実際の開発形態や開発体制などを考慮しながら前述の分類方法を適用し、辞書フォルダ構成を決定します。

辞書フォルダの分類方法と構成の例を次の図に示します。

図 13-1 辞書フォルダの分類と構成例



13.3 セキュリティ管理

辞書情報を保護するためのアクセス権、および排他制御について説明します。

13.3.1 辞書のアクセス権の設定

辞書の作成者や利用者というようにユーザを分類してアクセス権を設定すれば、辞書情報に対する操作を制限できるので、プロジェクトの大切な共有資源である辞書情報を保護し、整合性を保てます。ユーザの分類方法については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

(1) アクセス権を設定する単位

辞書に対するアクセス権は、次に示す単位で設定できます。

- 辞書フォルダ
- データ項目
- 業務ルール

(2) 辞書フォルダに設定するアクセス権

辞書フォルダに設定できるアクセス権は「読み取り」「書き込み」「作成」「削除」です。辞書フォルダのアクセス権は、リポジトリの環境構築ユティリティで変更できます。

(3) データ項目に設定するアクセス権

データ項目に設定できるアクセス権は「読み取り」「書き込み」「削除」です。データ項目のアクセス権は、クライアントのリポジトリブラウザで変更できます。

(4) 業務ルールに設定するアクセス権

業務ルールに設定できるアクセス権は「読み取り」「書き込み」「削除」です。業務ルールのアクセス権は、クライアントのリポジトリブラウザで変更できます。

(5) アクセス権設定例

辞書フォルダ、データ項目および業務ルールに設定するアクセス権とその結果を、例に示します。

(a) 設定するアクセス権

この例では、辞書を作成するグループと利用するグループを仮定し、それぞれがデータ項目および業務ルールに対し、次に示すようなアクセス権を持つよう設定します。

グループ	作成	変更	削除	参照
辞書作成者グループ				
辞書利用者グループ	x	x	x	

(凡例)

- ： 権利がある
- ： 自分で作成したデータ項目および業務ルールにだけ権利がある
- x： 権利がない

(b) 辞書フォルダのアクセス権

辞書フォルダのアクセス権を次のように設定します。

設定項目	設定内容
所有者名	system または辞書作成者グループのメンバ
グループ名	辞書作成者グループ
所有者のアクセス権	フルコントロール
グループのアクセス権	フルコントロール
全ユーザのアクセス権	読み取り

(c) データ項目および業務ルールのアクセス権

データ項目および業務ルールのアクセス権の初期値を示します。なお、アクセス権の初期値はカスタマイズできません。

設定項目	設定内容
所有者名	作成者
グループ名	作成者が属するグループ 属するグループが複数ある場合は、グループ登録ファイルの先頭に定義されたグループ
所有者のアクセス権	フルコントロール
グループのアクセス権	変更
全ユーザのアクセス権	読み取り

13.3.2 排他制御

複数のユーザが同じ辞書情報に対して操作しているとき、自動的に排他制御が働き、辞書情報の二重更新が防止されます。排他制御の機能は、辞書情報の整合性を保つことを目的としています。

- 排他制御の単位
辞書フォルダ、データ項目、および業務ルール単位の操作に排他制御が働きます。
- データ項目の階層単位の排他制御
データ項目は継承関係によって階層化されるため、上位データ項目の定義情報を変更すると、下位データ項目の定義情報にも影響を与えます。そこで、データ項目では、階層単位の操作についても排他制御が働きます。操作の対象にしているデータ項目が変更の影響を受ける、または影響を与える場合には、メッセージが表示されるため、留意してください。

14 分散開発環境

分散開発を支援するためのエクスポート・インポートの機能および運用方法について説明します。

14.1 分散開発環境とは

14.2 資源の配布

14.3 エクスポート・インポート時の設定

14.4 エクスポート・インポート時の注意事項

14.5 分散運用の流れ

14.1 分散開発環境とは

SEWB+ では、UNIX 上での大規模なシステム開発や分散環境に対応するため、複数のリポジトリを使用する複数サーバ運用を支援します。複数のリポジトリで運用をする場合、リポジトリ間で開発資源を受け渡す必要があります。

資源の受け渡しには、SEWB+ のエクスポート・インポート機能を利用します。エクスポート・インポート機能を利用して資源を受け渡すことを配布ともいいます。配布の運用方法によって、リポジトリ間の資源の整合性を確保できたり、複数のリポジトリ上にある開発資源を一つのリポジトリで統合管理できたりします。

14.1.1 配布の形態

配布の形態は、目的によって「レプリカ配布」「複製配布」「オリジナル配布」の三つに分けられます。なお、SEWB+ で推奨する配布形態は「レプリカ配布」です。

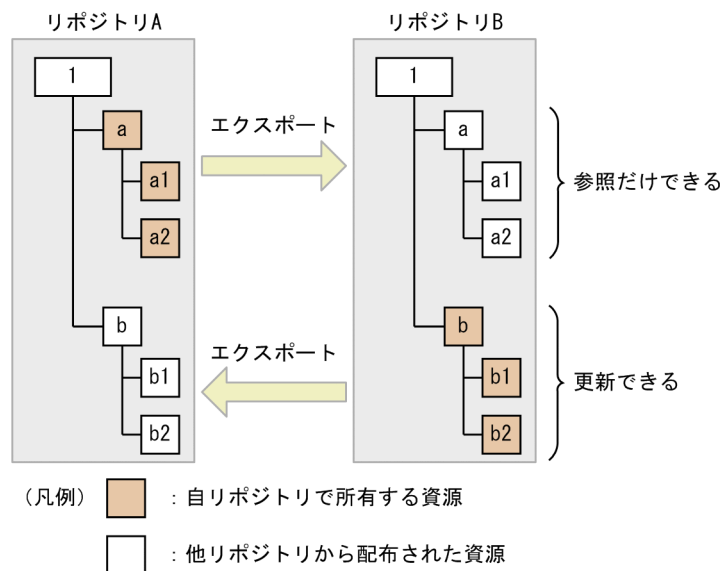
(1) レプリカ配布

複数のリポジトリを使用してシステム開発をする場合、それぞれのリポジトリがほかのリポジトリに参照目的で資源を配布できます。参照目的で資源を配布するため、他リポジトリでの資源の変更を抑制します。

自リポジトリにオリジナルの資源を所有し、他リポジトリに参照目的の資源をエクスポートすることをレプリカ配布といいます。

レプリカ配布を次の図に示します。

図 14-1 レプリカ配布

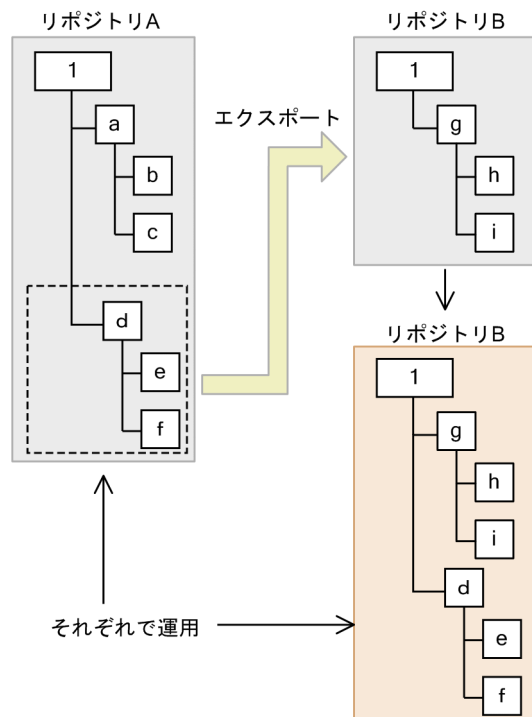


(2) 複製配布

あるリポジトリ上にある開発資源を、ほかのリポジトリに複製します。資源を複製した場合、複製先の資源と複製元の資源は、それぞれオリジナルとして別々に運用されます。

複製配布を次の図に示します。

図 14-2 複製配布



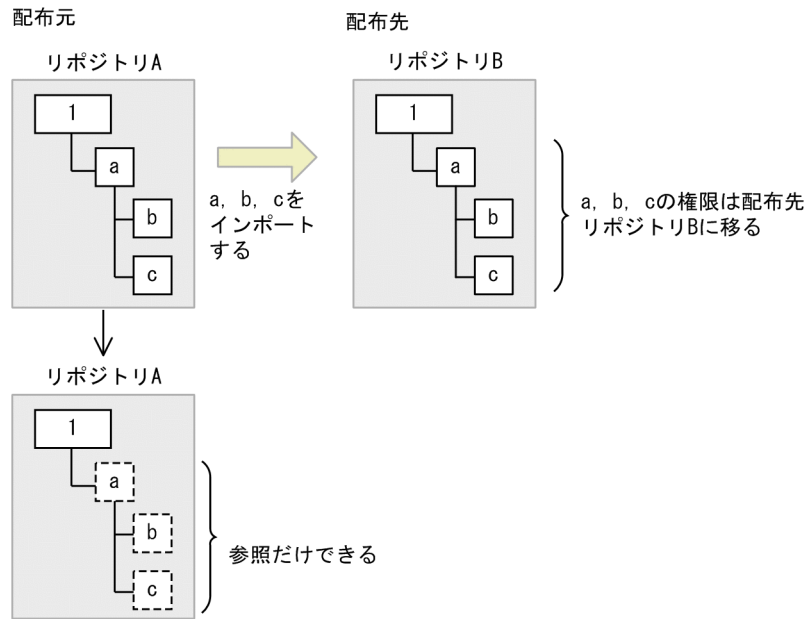
(3) オリジナル配布

あるリポジトリの資源を、ほかのリポジトリで修正したいときや、配布元のリポジトリに更新権を与えたくないときなどに適用します。また、個々のリポジトリで開発した資源を一つのリポジトリに統合するときにも利用できます。

この場合、配布元のリポジトリからオリジナル配布された資源への権限は配布先のリポジトリに移り、配布元のリポジトリでは参照目的でだけ使用できます。なお、配布先と配布元のリポジトリ間での相互配布はできません。

オリジナル配布を次の図に示します。

図 14-3 オリジナル配布



14.2 資源の配布

複数リポジトリ運用での資源の配布について説明します。

14.2.1 配布の対象

リポジトリ間で配布される資源を次に示します。

- ドキュメントフォルダ
- ドキュメント
- 辞書（データ項目辞書，業務ルール辞書）
- 環境構築情報（詳細は「14.2.4 環境構築情報のエクスポート」を参照のこと）

14.2.2 配布の目的

リポジトリ間で資源を配布する目的を次に示します。

- 自リポジトリでは自分の資源を管理し，開発に必要なそのほかの資源は他リポジトリから配布して運用するため（レプリカ配布）。
- リポジトリを複製して，新しく別のリポジトリを作るため（複製配布）。
- 自リポジトリの資源を，他リポジトリで修正するため（オリジナル配布）。

14.2.3 配布時の増分管理

リポジトリ間の資源配布には，エクスポート・インポートを利用します。ただし，毎回，すべての資源をエクスポート・インポートすると時間が掛かり，効率が下がります。SEWB+では，変更した箇所（増分）だけをエクスポート・インポートできます。

（1）ドキュメントの増分管理

レプリカ配布の場合に有効です。前回のエクスポート情報からの増分だけをエクスポートできます。

（2）辞書の増分管理

レプリカ配布の場合に有効です。前回のエクスポート情報からの増分だけをエクスポートできます。

14.2.4 環境構築情報のエクスポート

資源だけではなく，環境構築情報もエクスポートの対象になります。

（1）リポジトリの環境情報のエクスポート

環境構築ユーティリティで設定したリポジトリの環境情報をエクスポートできます。エクスポート・インポートできる環境構築ユーティリティの設定項目を次の表に示します。

表 14-1 エクスポート・インポートできる環境情報

環境構築ユーティリティの設定項目	エクスポート・インポート
システム管理者用のパスワード	×
同時ログイン可能ユーザ数最大値の設定	×
SEWB+ クライアントプログラムをターミナルサービス ¹ クライアントから使用するかどうかの設定	×

環境構築ユティリティの設定項目		エクスポート ・インポート
操作ログの設定		×
辞書フォルダ		2
データ項目と業務ルールの環境設定		3
辞書ユティリティタイプマッピングの設定		
ドキュメントの設定	ドキュメント、フォルダ、システムバージョンの初期属性	
	チェックアウト先コンピュータの IP アドレス	×
ドキュメント種別の設定		
ルートドキュメントフォルダ		2
バックアップの設定		×

(凡例)

○ : 対象にできる

× : 対象にできない

注 1

Windows Server 2008 R2 のリモートデスクトップサービスを示します。

注 2

エクスポート運用でドキュメントまたは辞書の資源をエクスポート対象に指定した場合に、ルートドキュメントフォルダまたは辞書フォルダの環境情報が対象となります。

注 3

命名ルール情報、および検査項目情報は対象外となります。

(2) 辞書の環境情報のエクスポート

辞書に関する環境構築情報として、次の情報をエクスポートできます。

- 辞書フォルダ
辞書資源をエクスポート対象にした場合にエクスポートされます。
- データ項目情報
データ項目や業務ルールに関する各種の環境構築情報がエクスポートされます。ただし、命名ルール情報や検査項目情報は対象になりません。
- タイプマッピング情報
ODBC 入力や、ERwin/ERX と連携した CSV 入力をするときのタイプの変換規則がエクスポートされます。

14.3 エクスポート・インポート時の設定

複数リポジトリ間で資源を相互配布する場合など、配布元と配布先の資源の整合性や、それぞれの資源間の関連を管理することが重要です。SEWB+ では、エクスポート・インポート時の設定によって、これらの問題に対処できます。

14.3.1 エクスポート時の設定

資源をエクスポートするときに、資源間に付けられている関連情報を一緒にエクスポートするかどうかを設定できます。

(1) 相互配布時のドキュメント関連情報

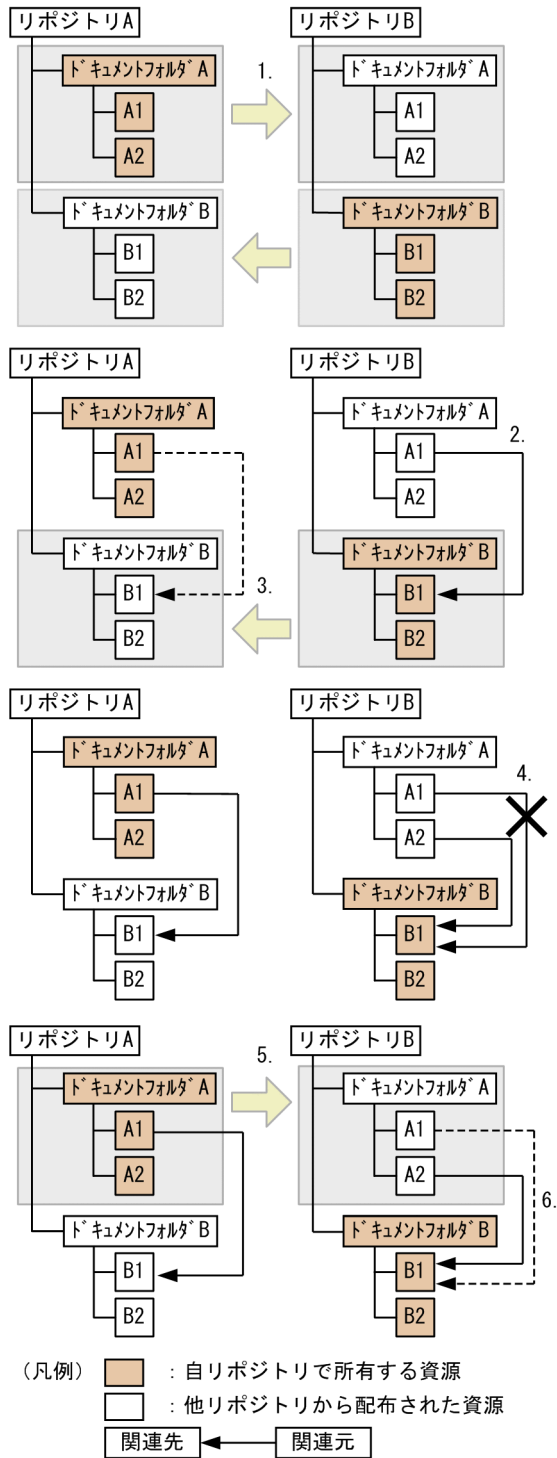
リポジトリ間でドキュメントを相互配布している場合、ドキュメントの関連情報をエクスポートするように設定できます。ただし、設定時には、次に示す「ドキュメント相互配布時の注意事項」などを考慮して設定してください。

(a) ドキュメント相互配布時の注意事項

リポジトリ間でドキュメントを相互配布する場合には、エクスポート運用の設定で「関連先となる開いた関連のみをエクスポートする」を選択してください。これは、インポート先で関連を削除したときに、インポートによって関連が復活するのを防ぐために必要です。

相互配布の関係にあるリポジトリ A とリポジトリ B があると仮定します。これらのリポジトリ間でのドキュメントの相互配布を例に、発生するおそれのある現象とその回避方法について説明します。ドキュメントの相互配布で起こることがある現象を次の図に示します。

図 14-4 ドキュメントの相互配布で起こることがある現象



1. 二つのリポジトリ間で、ドキュメントフォルダ A とドキュメントフォルダ B を相互配布する運用をしている。
2. リポジトリ B では、インポートしたドキュメントフォルダ A の A1 と自分が所有するドキュメントフォルダ B の B1 の間に関連を付ける。
このとき、A1 がテンプレート、B1 がプログラム定義と仮定します。テンプレートを関連元、プログラム定義を関連先として、参照の関連が付けられます。

3. リポジトリ B からリポジトリ A にドキュメントフォルダ B がエクスポートされる。
このとき、エクスポート時の設定はデフォルト状態と仮定します。デフォルトでは、すべての開いた関連がエクスポートされます。そのため、A1 と B1 の関連もエクスポートされます。
4. リポジトリ B 側で、A1 と B1 の関連を解除し、新たに A2 と B1 の間に関連を付ける。
A2 と B1 の関係は 2. と同じと仮定します。
5. リポジトリ A からリポジトリ B にドキュメントフォルダ A がエクスポートされる。
エクスポート時の設定はデフォルト状態と仮定します。そのため、A1 と B1 の関連もエクスポートされます。
6. A1 と B1 の間の関連が復活する。
4. で解除した A1 と B1 の関連が再び付けられます。これは、すべての関連をエクスポートするように設定されているため、5. のエクスポート時に、3. の時点でエクスポートされた関連情報が A1 の関連情報として再びリポジトリ B にエクスポートされるからです。

(b) 回避方法

(a) の現象は、エクスポートする関連情報を、関連方向をキーにして区別することで回避できます。例えば、エクスポート時に、開いた関連の関連先になるデータ項目の関連情報だけをエクスポートするように設定しておくこと、5. のエクスポート時に A1 の関連はエクスポートされなくなります。

注

この回避方法は、SEWB+ が関連を付ける規則（どの資源を関連元または関連先にするかを定める規則）に従って関連が付けられているときだけ有効です。したがって、ユーザ関連を付けるときは、SEWB+ の関連づけの規則に従うことをお勧めします。SEWB+ が付ける関連については、「11.1.5 関連の設定」およびマニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

(c) 開いた関連情報の運用

開いた関連情報をエクスポートすると、変更時の変更波及を調査するときに便利です。例えば、分散環境の複数リポジトリから参照目的で資源をインポートしていても、自分が所有する資源がそれらの関連先になっていれば、自分の資源を変更したときの相手資源への影響を調査できます。開いた関連情報をエクスポートしていないと、同じような状況のとき、リポジトリごとに調査しなければなりません。

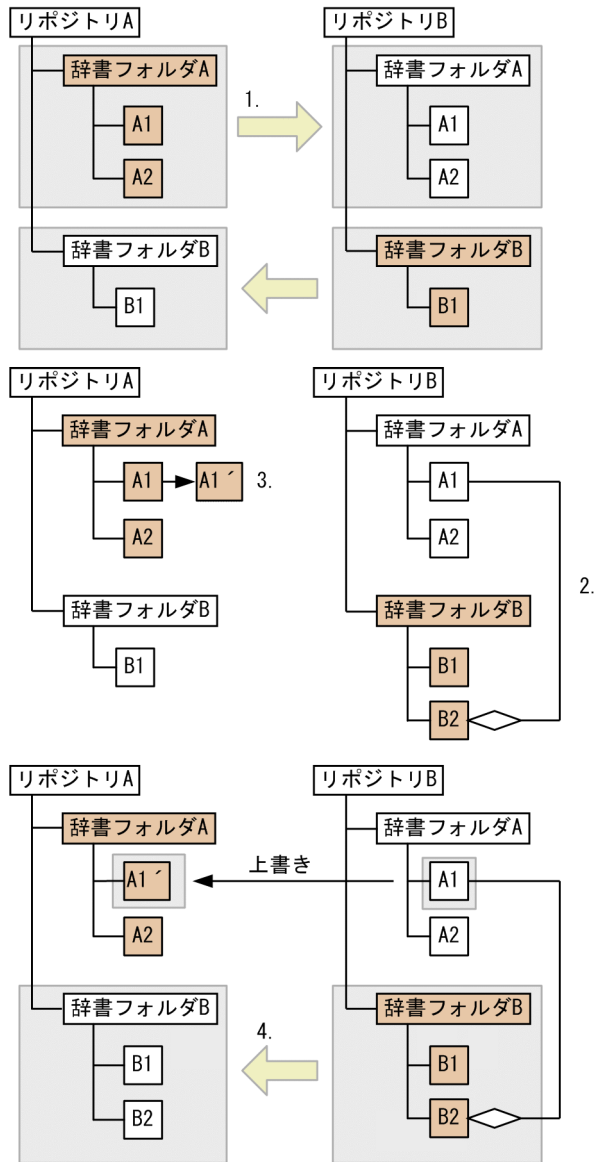
(2) 相互配布時の辞書のエクスポート範囲

リポジトリで複数の辞書を管理している場合、エクスポートする辞書フォルダのデータ項目が、エクスポート対象外の辞書のデータ項目と結合関係にあるとき、それらのデータ項目も一緒にエクスポートするかどうかを設定できます。設定時には、次に示す「辞書相互配布時の注意事項」などを考慮してください。

(a) 辞書相互配布時の注意事項

相互配布の関係にあるリポジトリ A とリポジトリ B があると仮定します。これらのリポジトリ間での辞書の相互配布を例に、発生するおそれのある現象とその回避方法について説明します。辞書の相互配布で起こることがある現象を次の図に示します。

図 14-5 辞書の相互配布で起こることがある現象



(凡例) : 自リポジトリで所有する資源
 : 他リポジトリから配布された資源
 : 結合関係

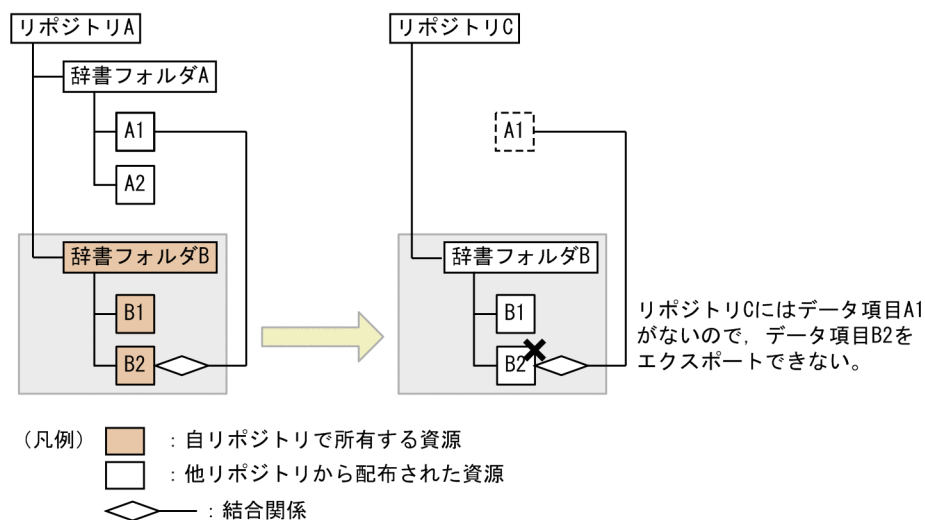
1. 二つのリポジトリ間で、辞書フォルダ A と辞書フォルダ B を相互に配布する運用をしている。リポジトリ A からリポジトリ B に辞書フォルダ A がエクスポートされる。
 このとき、エクスポート時の設定はデフォルト状態と仮定します。デフォルトでは、結合関係にあるすべてのデータ項目がエクスポートされます。
2. リポジトリ B では、インポートした辞書フォルダ A の A1 を使用して、自分が所有する辞書フォルダ B に結合項目 B2 を追加する。
3. リポジトリ A では、辞書フォルダ A の A1 を変更する。
4. リポジトリ B からリポジトリ A に辞書フォルダ B がエクスポートされる。
 設定は 1. と同じと仮定します。

このとき、B2の構成項目であるA1も一緒にエクスポートされるため、リポジトリA側で変更した内容(A')が上書きされてしまいます。

(b) 回避方法

このようになることが事前にわかっている場合は、エクスポート時の設定で、ほかの辞書フォルダのデータ項目と一緒にエクスポートする設定を解除しておきます。ただし、このように設定すると、ほかの辞書フォルダにある構成項目がインポート先のリポジトリにない場合にエラーとなるので注意が必要です。

例えば、前述の運用例で、リポジトリBの辞書フォルダの結合項目B2は辞書フォルダAのA1を構成項目にしています。この場合に、リポジトリBが別のリポジトリであるリポジトリCに辞書フォルダBをエクスポートしようとする、辞書フォルダCにはデータ項目A1がないのでエラーになります。



このような現象を回避するためには、分散環境のすべてのリポジトリのフォルダ構成を同じにしてください。または、参照されるデータ項目をドメインとして一つの辞書フォルダにまとめて共用してください。

ドメインを格納する辞書フォルダについては、「13.2.2 データ項目の分類方法」を参照してください。

14.3.2 インポート時の設定

相互配布のリポジトリ間でレプリカ配布をしたい場合、インポート先では、エクスポートされた資源に対し、新たに読み取り専用のアクセス権を設定してください。

14.3.3 エクスポート・インポート運用の設定例

エクスポート・インポートの設定例を次に示します。

(1) エクスポートの設定と実行

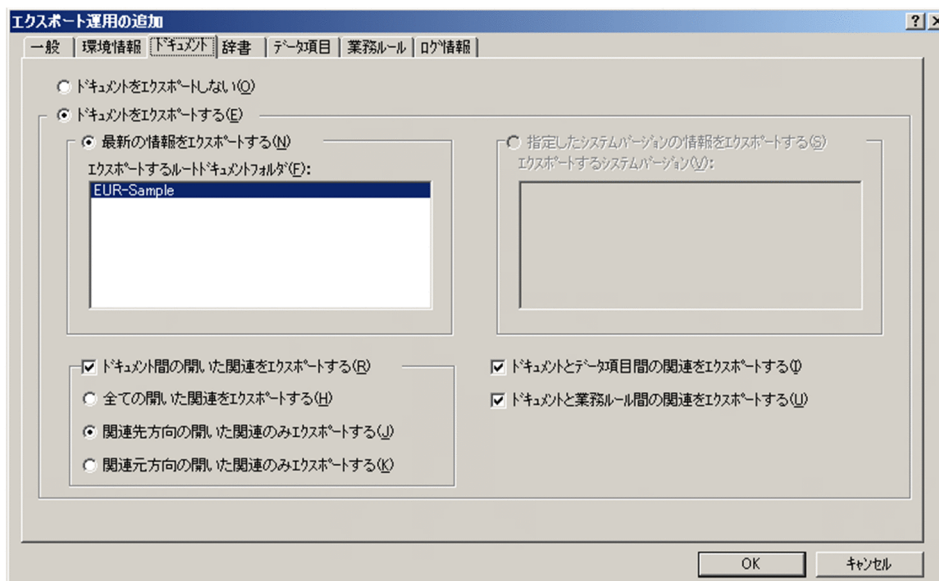
資源を配布する側は、次のように設定して、エクスポートを実行します。

(a) エクスポートの設定

1. [一般] タブに次の設定をする。

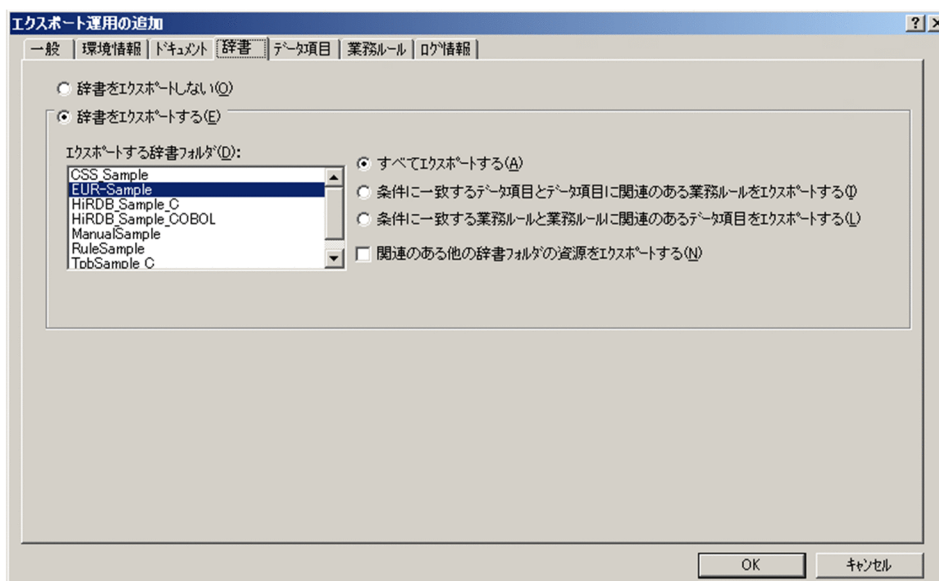
項目	設定内容
運用名	任意の運用名を設定する。
コメント	任意に指定する。
エクスポートできない資源があっても続行する	選択しない(ドキュメントをチェックアウト中などでエクスポートできない場合中止とするため)
エクスポートファイル	出力するエクスポートファイルのファイル名として、任意のファイル名を指定する。
エクスポートファイルの最大サイズ	1本の媒体に格納可能なファイルサイズ(単位:MB)を指定する(例えば、230MB格納可能なMOを使う場合は「230」と指定する)。
既存のエクスポートファイルを上書きする	選択する(設定したエクスポート運用で繰り返し配布をするため、既存ファイルがあれば上書きする)

2. [ドキュメント] タブに次の設定をする。



項目	設定内容
ドキュメントをエクスポートする	選択する。
最新の情報をエクスポートする	選択する。
エクスポートするルートドキュメントフォルダ	エクスポートの対象とするフォルダを選択する。
ドキュメント間の開いた関連をエクスポートする	「関連先方向の開いた関連のみエクスポートする」を選択する（削除した関連が復活することを防ぐため）。
ドキュメントとデータ項目間の関連をエクスポートする	選択する（CONSTRUCTION データ定義ファイルなどはデータ項目と関連を持つため）。
ドキュメントと業務ルール間の関連をエクスポートする	選択する（CONSTRUCTION プログラム定義ファイルは業務ルールと関連を持つため）。

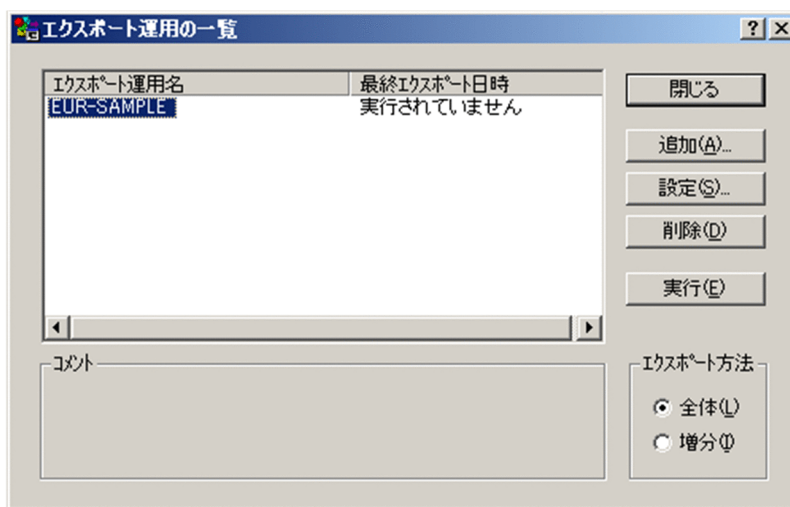
3. [辞書] タブに次の設定をする。



項目	設定内容
辞書をエクスポートする	選択する。
エクスポートする辞書フォルダ	エクスポートする辞書フォルダを選択する。
すべてエクスポートする	選択する（この場合、[データ項目]タブおよび[業務ルール]タブは設定不要）。
関連のある他の辞書フォルダの資源をエクスポートする	設定を解除する（ほかの辞書フォルダの構成項目を配布しないようにするため）。

(b) エクスポートの実行

上記のように設定した状態で [エクスポート運用の一覧] ダイアログに次のように設定します。[実行] ボタンをクリックしてエクスポートを実行します。



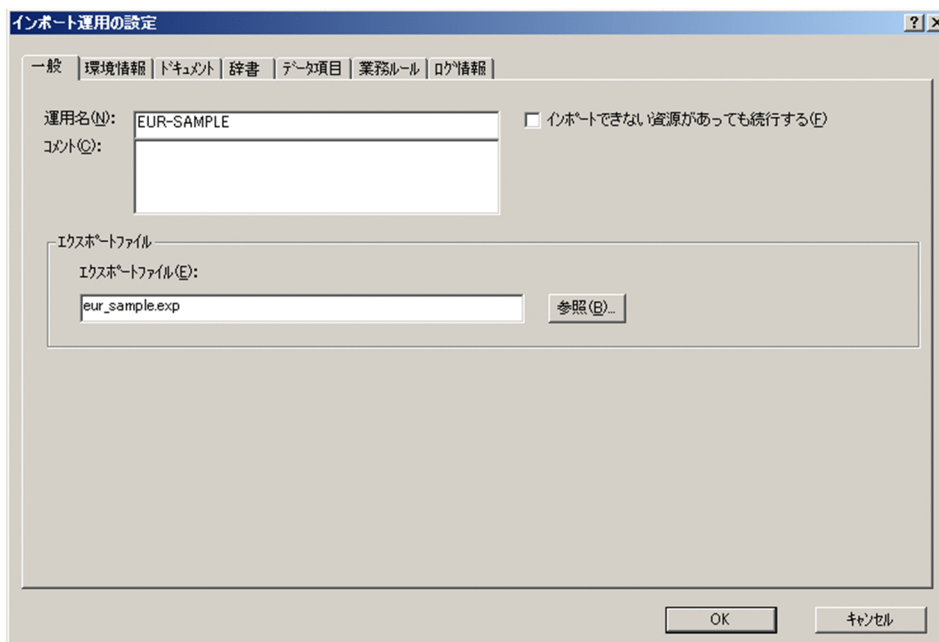
項目	設定内容
エクスポート運用名	[一般] タブで設定した運用名を選択する。
エクスポート方法	<ul style="list-style-type: none"> • 1回目の配布時：「全体」 • 2回目以降の配布時：「増分」 • 開発終了時：「全体」

(2) インポートの設定と実行

資源を受け取る側は、次のように設定して、インポートを実行します。

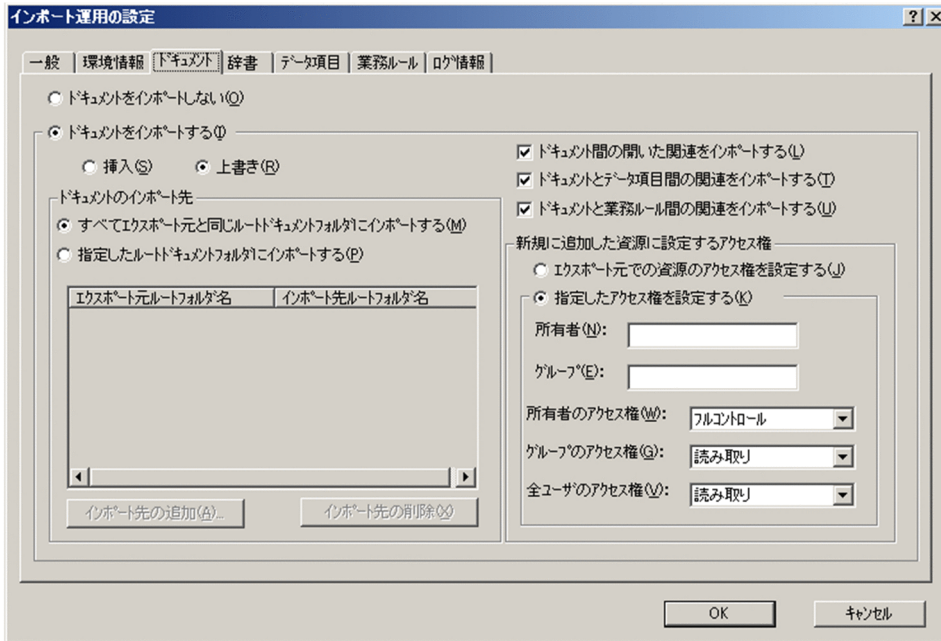
(a) インポートの設定

1. [一般] タブを次のように設定する。



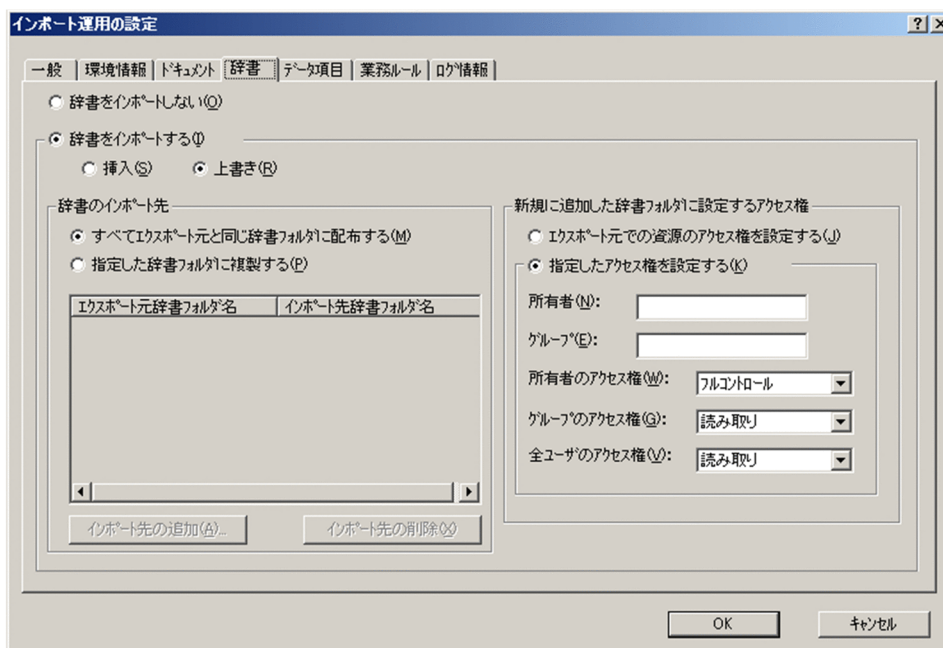
項目	設定内容
運用名	任意の運用名を設定する。
コメント	任意に指定する。
インポートできない資源があっても続行する	選択しない(確実にインポートされなかった場合はエラーとするため)。
エクスポートファイル	エクスポート側の運用で指定したエクスポートファイル名を設定する。

2. [ドキュメント] タブを次のように設定する。



項目	設定内容
ドキュメントをインポートする	選択する。
挿入 / 上書き	「上書き」を選択する。
ドキュメントのインポート先	「すべてエクスポート元と同じルートドキュメントフォルダにインポートする」を選択する（リポジトリサーバ間でフォルダ名を一致させるため）。
ドキュメント間の開いた関連をインポートする	選択する（配布される資源とそれ以外の資源との関連を含めて配布する）。
ドキュメントとデータ項目間の関連をインポートする	選択する（CONSTRUCTION データ定義ファイルなどはデータ項目と関連を持つため）。
ドキュメントと業務ルール間の関連をインポートする	選択する（CONSTRUCTION プログラム定義ファイルは業務ルールと関連を持つため）。
新規に追加した資源に設定するアクセス権	「指定したアクセス権を設定する」を選択する（下記のアクセス権とするため）。
所有者	指定しない。デフォルトの「インポート実行者」（システム管理者）が仮定される。
グループ	指定しない。デフォルトの「インポート実行者」が所属するグループが仮定される。
所有者のアクセス権	フルコントロールを選択する（システム管理者はフルコントロールであるため）。
グループのアクセス権	読み取りを選択する（配布先では更新できないようにする）。
全ユーザのアクセス権	読み取りを選択する（配布先では更新できないようにする）。

3. [辞書] タブを次のように設定する。



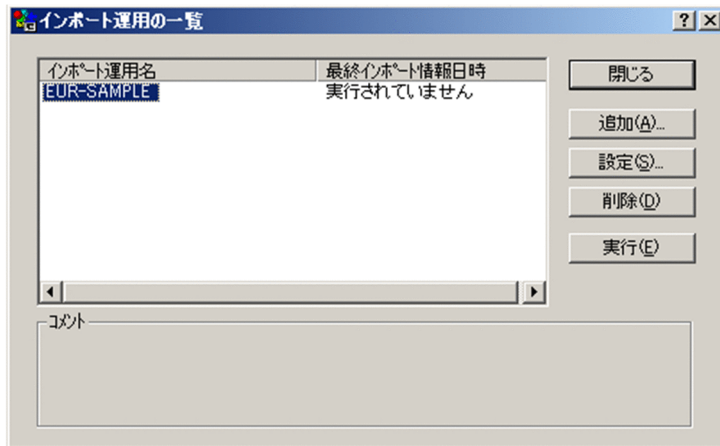
項目	設定内容
辞書をインポートする	選択する。
挿入 / 上書き	「上書き」を選択する（同一データ項目に上書きするため）。
辞書のインポート先	「すべてエクスポート元と同じ辞書フォルダに配布する」を選択する（リポジトリサーバ間でフォルダ名を一致させるため）。
新規に追加した辞書フォルダに設定するアクセス権	[ドキュメント] タブと同様に設定する。

4. [データ項目] タブおよび [業務ルール] タブを次のように設定する。

項目	設定内容
新規に追加した資源に設定するアクセス権	[ドキュメント] タブと同様に設定する。

(b) インポートの実行

これまで説明した内容を設定した状態で [インポート運用の一覧] ダイアログに次のように設定します。
[実行] ボタンをクリックしてインポートを実行します。



項目	設定内容
インポート運用名	[一般] タブで設定した運用名を選択する。

14.4 エクスポート・インポート時の注意事項

エクスポート・インポートでは、次の項目に注意してください。

(1) ノード ID

エクスポート・インポートでは、複数のリポジトリ間で資源をやり取りします。このため、指定するノード ID は、複数のリポジトリのデータベース間で重複しない値にしてください。

デフォルト値のノード ID で複数のデータベースを初期化すると、ノード ID が重複するので注意してください。

注

ノード ID は DB を初期化するときに使用します。

(2) 資源の削除

インポートを実行すると、エクスポートされた資源がインポート先に新規に追加、またはエクスポートされている資源に対応するインポート先の資源が置換されます。しかし、インポートの実行でインポート先の資源が削除されることはありません。このため、次のような運用をお勧めします。

- インポート先に不要な資源がある場合、インポートを実行しても不要な資源は削除されません。インポート先に不要な資源がある状態でインポートを実行すると、インポートの実行後は不要な資源とインポートした資源が混在することになります。このような場合は、インポートを実行する前にインポート先の不要な資源を削除し、その後インポートを実行してください。
- エクスポート元のリポジトリの資源をエクスポート・インポート機能を使用してほかのリポジトリに複製する運用を継続的に行っている場合、エクスポート・インポートを実行してもエクスポート元のリポジトリで削除された資源に対応するインポート先の資源は削除されません。このような場合は、エクスポート元で削除された資源に対応するインポート先の資源を削除してください。

(3) Windows Server の at コマンドなどの利用

エクスポート・インポート時には、リポジトリにアクセスしないことをお勧めします。そのため、時間の掛かるエクスポート・インポートは夜間や休日に行い、at コマンドなどを使用して自動的に実行させてください。

(4) 資源の配布

資源をほかのリポジトリに配布したあとで、資源を再配布する場合は、インポート時の設定で「上書き」を設定してください。インポート時の設定で「挿入」を選択すると、先に配布した資源には上書きされません。

(5) 実行中の中止

インポートを実行中に中止した場合、中止する前までの資源はインポート先に登録されているので、エクスポート元のリポジトリとインポート先のリポジトリで矛盾が生じてしまいます。インポート先のリポジトリを回復するためには、中止したインポートを再実行してインポートを完了させる必要があります。

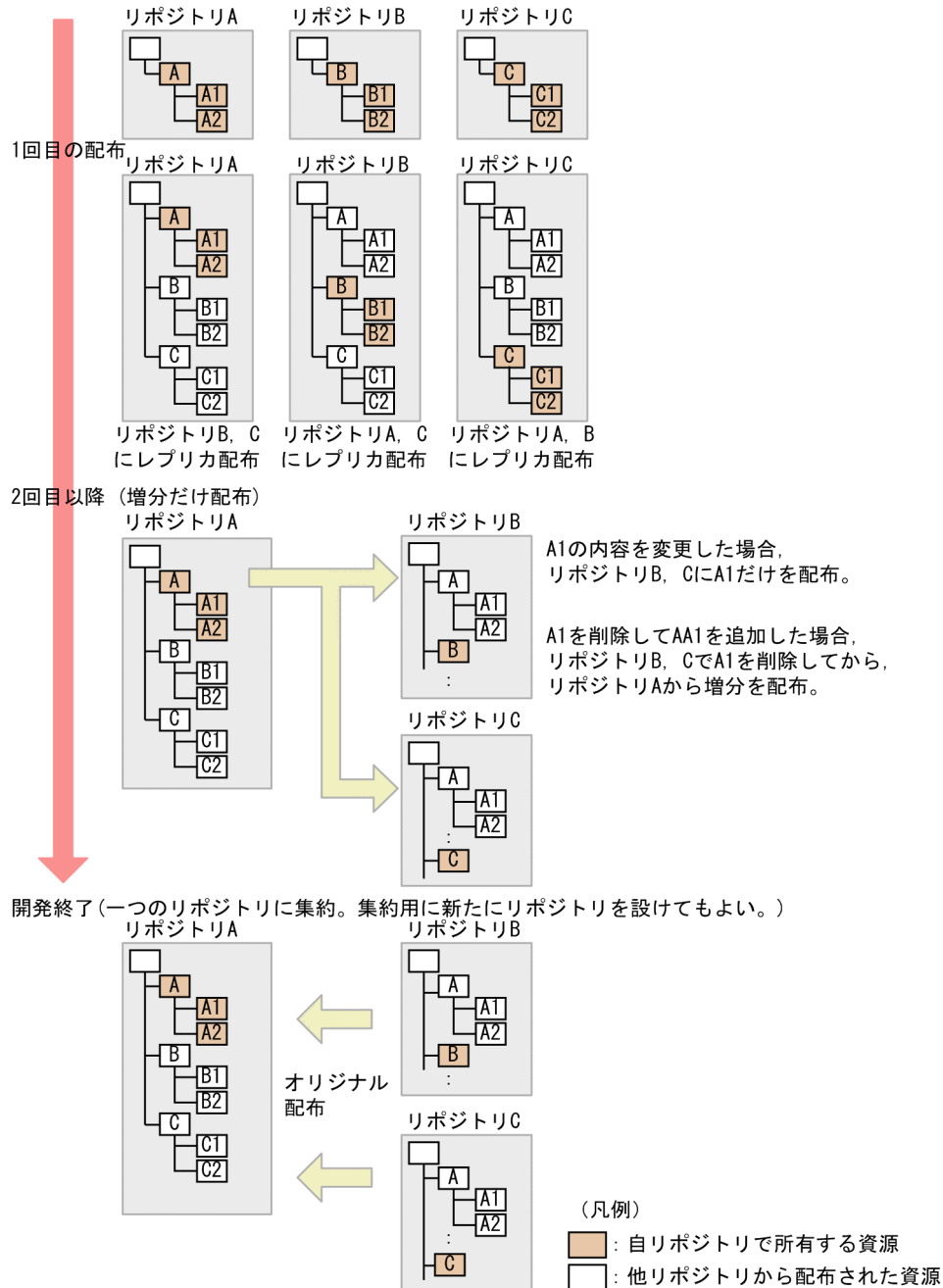
エクスポートを実行中に中止した場合、作成途中のエクスポートファイルは削除されます。ただし、ファイルサイズを指定してエクスポートファイルの分割作成をする指定をしていた場合、作成済みのエクスポートファイルは削除されません。

エクスポート・インポートの中止手順については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

14.5 分散運用の流れ

分散運用の流れを次の図に示します。

図 14-6 分散運用の流れ



15 SEWB+ のノウハウ

この章では、SEWB+ を利用したシステム開発のノウハウについて説明します。

15.1 開発ライフサイクルとリポジトリ管理

15.2 エクスポート・インポートの自動運用

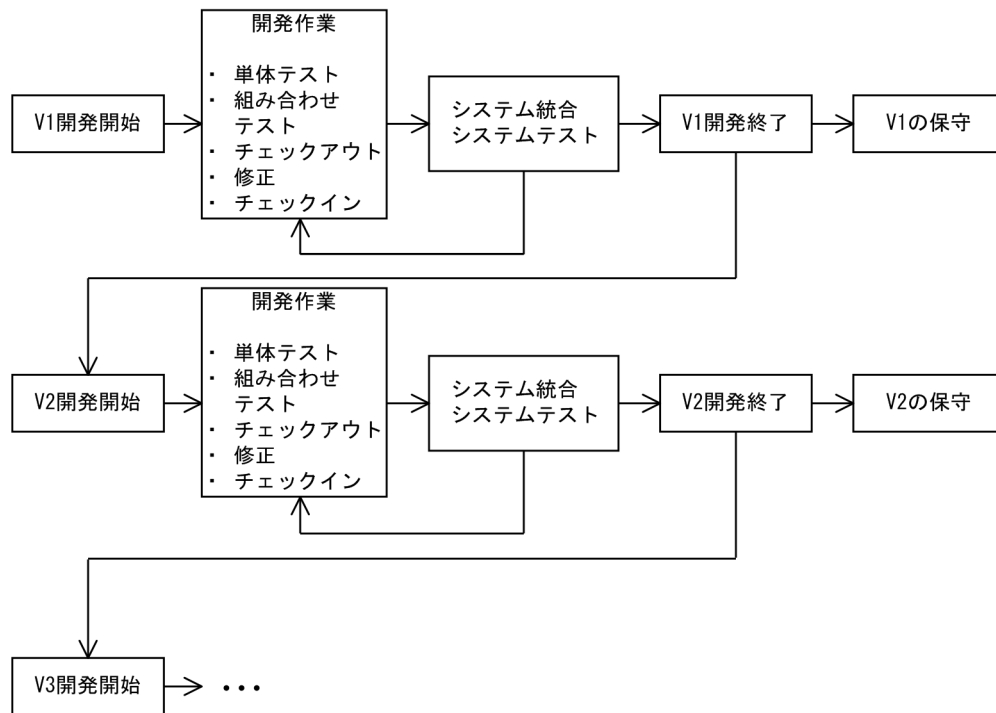
15.1 開発ライフサイクルとリポジトリ管理

ここでは、SEWB+ を利用したシステム開発のライフサイクルと、それに伴うリポジトリ管理について説明します。

15.1.1 開発ライフサイクル

SEWB+ を利用して大規模な開発をするときには、次の図に示すライフサイクルで開発資源を管理していくことになります。

図 15-1 開発ライフサイクル



(凡例)

- V1 : バージョン1
- V2 : バージョン2
- V3 : バージョン3

バージョン 1 の開発作業では、機能別に作成した個々のプログラムをリポジトリにチェックインして登録します。これらのプログラムをすべてリポジトリ内に登録したあとで、実行環境上で動作するシステム構成にまとめ上げます。これをシステム統合 (System Integration : SI) といいます。

システム統合で作成したシステムは、実機で動作確認をします。これを実機テストといいます。実機テストで動作に不具合が発見された場合には、再び開発作業に戻り、リポジトリに格納されている資源をチェックアウトして修正を加えます。修正が終わったら、再びチェックインします。

このようにしてシステム統合と実機テストを繰り返したあと、実機テストで不具合が発生しなくなれば、バージョン 1 の開発作業は終了します。

バージョン 1 で開発した生成物は、次のバージョン (バージョン 2) の開発の基盤となります。バージョン 2 以降は、開発作業とシステム統合・実機テストを繰り返して開発を進めていきます。

また、バージョン 1 の生成物は、リリースしたバージョン 1 のシステムに障害が発生した場合の調査およ

び対策のために使用します。このため、バージョン1のすべての資源を保管しておく必要があります。

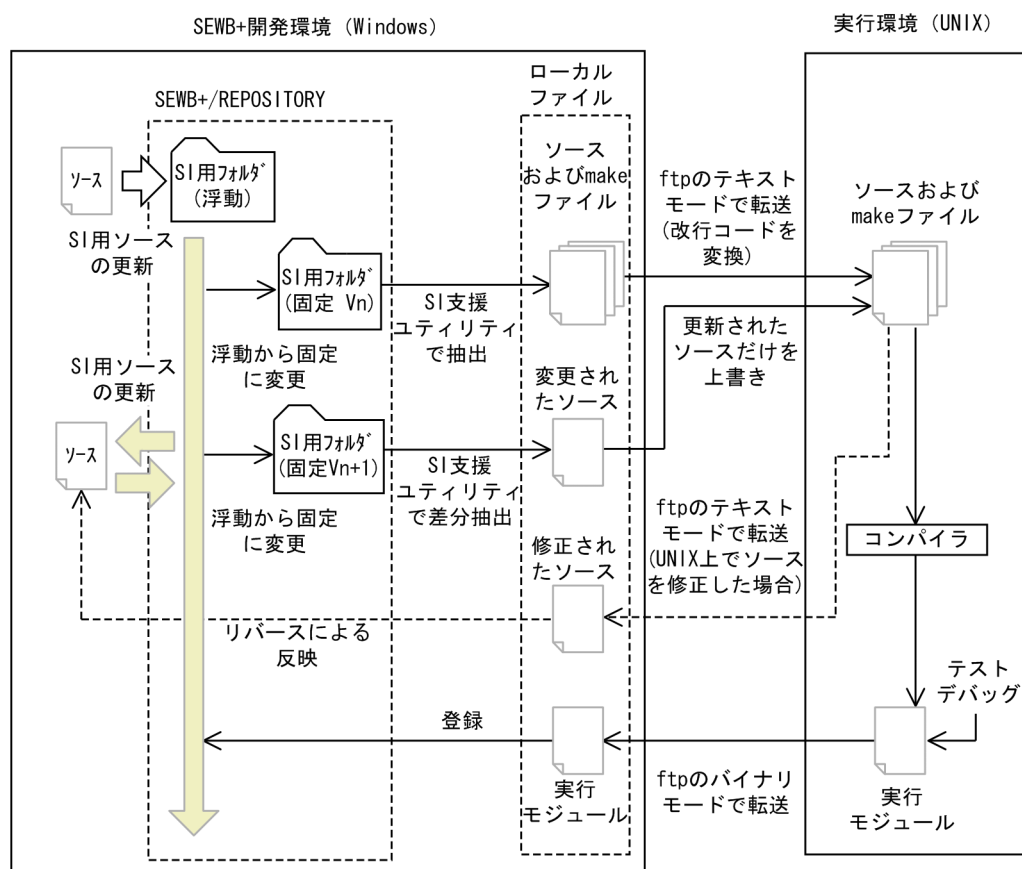
システムの一とおりの機能が開発されたあと、実機で動作確認をするときの作業の流れを次の項で説明します。

15.1.2 UNIX 環境でのシステム統合・実機テスト時の作業の流れ

実行環境が UNIX の場合、Windows 環境上の SEWB+ で開発した成果物を UNIX 側に転送してからコンパイルおよび動作確認をする必要があります。

UNIX 上で動作確認をする場合の作業の流れを次の図に示します。

図 15-2 UNIX 環境での動作確認



作業手順を次に示します。

1. ソースを含めた開発資源をリポジトリにドキュメント登録する
 これによって、資源をチェックインされた状態にしておきます。それらの資源のうち、システムを構成するためのソースをシステム統合用ソースフォルダにまとめ、浮動システムバージョンとして登録しておきます。
 開発作業が完了し、システム統合を開始する段階で、浮動システムバージョンを固定システムバージョンに変更します。
2. 固定システムバージョンに登録されている資源を、SI支援ユーティリティを使ってローカルファイル上に取り出す
3. 取り出したファイル群を実行環境に転送する

この時に、ftp のテキストモードで転送することによって改行コードが変換されます。

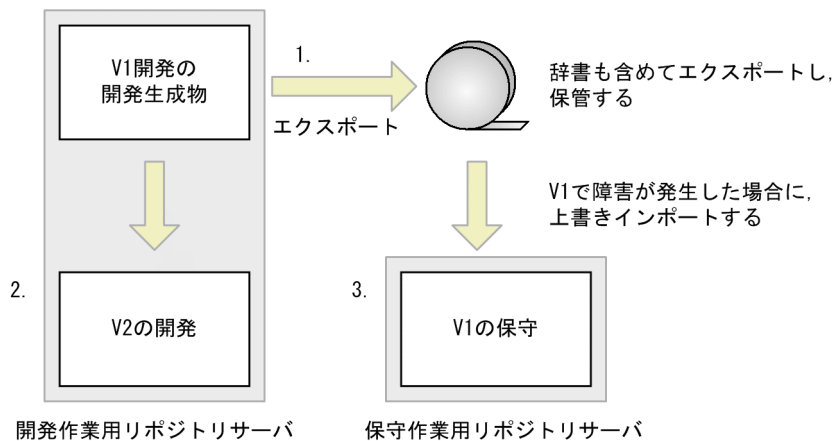
4. コンパイルおよびリンクを実行して、実行モジュールを作成する
5. テストおよびデバッグを実行する
実機テストで不具合があった場合には、リポジトリに格納された資源を修正します。リポジトリの資源に対してチェックイン・チェックアウトをすると、浮動システムバージョンには、常に最新のドキュメントのバージョンが反映されます。
6. 資源を再び固定システムバージョンに変更する
SI 支援ユーティリティの差分機能を利用して、前回、固定システムバージョンに変更した時以降に変更があったドキュメントだけを取り出します。これを転送して UNIX 上のコンパイル環境に上書きすることによって、make コマンドを使って、修正のあったソースだけを再コンパイルできます。

15.1.3 リポジトリへの資源の反映

実機テストで不良が見付からなくなればシステムは完成です。完成したシステムは、次のバージョンの基盤となります。また、運用後の保守にも使用します。このため、各種資源を保管しておく必要があります。

システム完成後の資源の保管手順を次の図に示します。

図 15-3 システム完成後の資源の保管手順



(凡例)

V1 : バージョン1
V2 : バージョン2

1. 固定システムバージョンにしたバージョン 1 の開発生成物を、データ項目辞書も含めてエクスポートし、外部記憶媒体に保管する
2. バージョン 1 の資源は、開発作業用リポジトリサーバ上でバージョン 2 の開発作業の基盤として利用する
3. リリースしたバージョン 1 のシステムに障害が発生した場合、バージョン 2 の開発作業を進めている開発作業用リポジトリとは別に、保守作業用リポジトリサーバを用意する。そこにレプリカ配布で辞書をインポートし、保守作業を実施する

SEWB+ では、ドキュメントのバージョン管理はしますが、辞書フォルダのバージョン管理はしません。このため、バージョン 1 の生成物を基盤にしてバージョン 2 の開発作業をする場合に、バージョン 2 の開発作業の中でデータ項目辞書に変更を加えると、同じデータ項目を参照しているバージョン 1 の開発生成物も影響を受けてしまいます。その場合、保守作業をしようとしたときに、同じ内容のプログラムを生成できないことがあります。

これを避けるには、バージョン1の開発生成物を、辞書を含めてエクスポートして保管します。保守作業が必要になったときには、バージョン2開発とは別のリポジトリサーバにインポートします。

15.2 エクスポート・インポートの自動運用

エクスポート・インポートの実行中、対象資源は排他制御されません。そのため、エクスポート・インポートを実行している資源に対してほかの操作が実行された場合、操作の内容によってはエラーが発生することがあります。このため、エクスポート・インポートは、夜間に自動運用することをお勧めします。

Windows Server の at コマンドを使用すると、指定した時刻にエクスポート・インポートを実行できます。このとき、誤って資源を操作しないようにするために、最大ログインユーザ数変更コマンド (RPMaXusr) をエクスポート・インポート用のバッチファイルの中に指定して、最大ログインユーザ数を 1 にします。また、自動実行中にダイアログが表示されないようにするために、エクスポートのコマンドに /nomsg オプションを指定します。

at コマンドでエクスポートを実行するバッチファイルの例を次に示します。

<pre> REM *** PATH 環境変数の設定 *** SET PATH=%PATH%;“c:¥Program Files¥Hitachi¥Sewb¥Repository¥bin” ※ REM *** SEWB+サービスの停止 *** NET STOP SEWB REM *** 最大ログインユーザ数を 1 に変更する *** RPMaXusr 1 IF %ERRORLEVEL% EQU -1 goto :OMIT SET SAVEMAXUSR=%ERRORLEVEL% REM *** SEWB+サービスの起動 *** NET START SEWB REM *** エクスポートの実行 *** RPexport 自動エクスポートの運用定義 /nomsg REM *** SEWB+サービスの停止 *** NET STOP SEWB REM *** 変更前の最大ログインユーザ数の回復 *** RPMaXusr %SAVEMAXUSR% REM *** SEWB+サービスの起動 *** NET START SEWB OMIT : PAUSE </pre>	<p>← 最大ログインユーザ数を 1 に変更する</p> <p>← 実行中にダイアログが表示されないように、/nomsgを指定する</p>
--	---

注※ 64ビット版のWindowsの場合は、次のようになります。
SET PATH=%PATH%;“c:¥Program Files (x86)¥Hitachi¥Sewb¥Repository¥bin”

付録

付録 A このマニュアルの参考情報

付録 B 用語解説

付録 A このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 A.1 関連マニュアル

SEWB+ のマニュアル体系を次に示します。

SEWB+ 概説	(概) (3020-3-B80)	
リポジトリ管理		
SEWB+/REPOSITORY 運用ガイド	(解)(手) (3020-3-B81)	
SEWB+/REPOSITORY 辞書設計ガイド	(手) (3020-3-B82)	
Groupmax Object Server Version 6 システム管理者ガイド	(解)(手) (3020-3-B56)	
オブジェクト指向分析・設計		
SEWB+ オブジェクト指向分析・設計支援 使用の手引	(手)(操) (3020-3-581)	
クライアントサーバシステム設計支援		
SEWB+/OLTP DEFINER ユーザーズガイド	(手)(操) (3020-3-909)	
アプリケーション開発		
SEWB+/CONSTRUCTION アプリケーション開発ガイド	(手)(文) (3020-3-B83)	
SEWB+/RECORD DEFINER 操作ガイド	(手)(操) (3020-3-B84)	
SEWB+/REPORT MANAGER ドキュメント作成支援 使用の手引	(手) (3020-3-720)	
SEWB+/CODE-DESIGN コード設計支援 使用の手引	(手) (3020-3-721)	
SEWB+ パッチシステム向けアプリケーションフレームワーク・部品 使用の手引	(手) (3020-3-711)	
SEWB+/STANDARD-SUBROUTINE 標準サブルーチン 使用の手引	(手) (3020-3-725)	
SEWB+ COBOL 構造図エディタ 使用の手引	(手)(操) (3020-3-811)	
		<記号>
		(概) : 概説書
		(解) : 解説書
		(手) : 手引書
		(文) : 文法書
		(操) : 操作書

関連マニュアルを次に示します。必要に応じてお読みください。

SEWB+ 基本開発環境セット

- SEWB+/REPOSITORY 運用ガイド (3020-3-B81)
- SEWB+/REPOSITORY 辞書設計ガイド (3020-3-B82)
- SEWB+/CONSTRUCTION アプリケーション開発ガイド (3020-3-B83)
- SEWB+/RECORD DEFINER 操作ガイド (3020-3-B84)

SEWB+ 関連製品

- SEWB+/OLTP DEFINER ユーザーズガイド (3020-3-909)

COBOL

- COBOL2002 操作ガイド (3020-3-D41)
- COBOL2002 操作ガイド (3020-3-D47)

WorkCoordinator

WorkCoordinator Definer Version 3 ユーザーズガイド (3020-3-987)

EUR

- uCosminexus EUR 概説 (3020-7-531) ¹
- uCosminexus EUR 帳票設計 (3020-7-533) ¹
- uCosminexus EUR 帳票設計 (EUR Form 帳票)(3020-7-534) ¹
- uCosminexus EUR クライアント帳票出力 (3020-7-535) ¹
- uCosminexus EUR サーバ帳票出力 (3020-7-536) ¹
- uCosminexus EUR メッセージ (3020-7-537) ¹
- uCosminexus EUR システム設計ガイド (3020-7-702) ²
- uCosminexus EUR システム構築ガイド (3020-7-703) ²
- uCosminexus EUR 帳票作成 画面リファレンス (3020-7-705) ²
- uCosminexus EUR 帳票出力 機能解説 EUR Server 編 (3020-7-708) ²
- uCosminexus EUR 帳票出力 リファレンス EUR Server 編 (3020-7-710) ²
- uCosminexus EUR データベース連携ガイド (3020-7-713) ²
- uCosminexus EUR メッセージ (3020-7-714) ²

XMAP3

- 画面・帳票サポートシステム XMAP3 プログラミングガイド 画面編 (3020-7-583) ³
- 画面・帳票サポートシステム XMAP3 プログラミングガイド 帳票編 (3020-7-584) ³
- 画面・帳票サポートシステム XMAP3 プログラミングガイド 帳票編 (3020-7-593) ⁴
- 画面・帳票サポートシステム XMAP3 プログラミングガイド 画面編 (3020-7-594) ⁴
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド (3020-7-512) ⁵
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド (3020-7-513) ⁵

TPBroker

トランザクショナル分散オブジェクト基盤 TPBroker ユーザーズガイド (3000-3-775)

OpenTP1

- OpenTP1 Version 7 分散トランザクション処理機能 OpenTP1 解説 (3000-3-D50)
- OpenTP1 Version 7 分散トランザクション処理機能 OpenTP1 システム定義 (3000-3-D52)
- OpenTP1 Version 7 分散トランザクション処理機能 OpenTP1 運用と操作 (3000-3-D53)
- OpenTP1 インターネットゲートウェイ機能 TP1/Web 使用の手引 (3000-3-D62)

- 注 1
EUR Version 8 の場合にお読みください。
- 注 2
EUR Version 9 の場合にお読みください。
- 注 3
XMAP3 Version 3 の場合にお読みください。
- 注 4
XMAP3 Version 4 の場合にお読みください。
- 注 5
XMAP3 Version 5 の場合にお読みください。

付録 A.2 このマニュアルでの表記

このマニュアルでは、製品名および機能名を次のように表記しています。

表記		正式名称
EUR		EUR : イーユーアール Professional Edition
		uCosminexus EUR : イーユーアール Designer
		uCosminexus EUR : イーユーアール Developer
Object Server	Groupmax Object Server	Groupmax Object Server Version 6
	Groupmax High-end Object Server	Groupmax High-end Object Server Version 6
SEWB+/CONSTRUCTION		SEWB+ 基本開発環境のプログラム構築支援機能
SEWB+/RECORD DEFINER		SEWB+ 基本開発環境のレコード設計支援機能
SEWB+/REPOSITORY	SEWB+/REPOSITORY	SEWB+ 基本開発環境のリポジトリ管理機能
	SEWB+/REPOSITORY-BROWSER またはリポジトリブラウザ	SEWB+ 基本開発環境のリポジトリブラウザ機能

- このマニュアルではパターンテンプレートをテンプレート、部品テンプレートを部品と表記しています。ただし、仕様書名にはそのまま「パターンテンプレート」「部品テンプレート」を使用しています。

付録 A.3 KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ $1,024$ バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

付録 B 用語解説

(英字)

C/S システム

クライアントサーバシステムの略称です。サービスを提供するサーバと、サービスを要求するクライアントから構成されます。一つの処理をネットワーク上のクライアントとサーバに分散させる業務形態です。

C/S 処理フロー設計

システム設計の機能分散処理の設計時の作業です。機能の分散化設計で C/S システム上のどこに配置するか決定された各機能を詳細化し、業務処理ごとに操作およびシステム処理の流れとして設定します。

DOA (Data Oriented Approach)

データ中心アプローチのことです。詳細は、用語解説の「データ中心アプローチ」を参照してください。

DOA 手法

データ中心アプローチの考え方に沿ったシステム開発の手法です。

ER 図 (Entity Relationship 図)

データ項目などの実態を表すエンティティとエンティティ属性、およびエンティティ間の関連であるリレーションシップで構成されるデータモデルです。

EUR

SEWB+/REPOSITORY と連携し、レポートの設計から印刷までを支援するツールです。

Object Server

Groupmax Object Server および Groupmax High-end Object Server は、多様な構造と関連性を持った情報をオブジェクトとして格納するプログラムです。また、それらのオブジェクトを管理するための高度な管理機能をそなえています。このマニュアルでは、両方のプログラムに共通の説明をする場合には、これらのプログラムを総称して Object Server と呼んでいます。

SEWB+/CONSTRUCTION

SEWB+ 基本開発環境のプログラム構築支援機能のことです。

特に、TP モニタやジョブ運用管理ツールなどを利用する大規模な C/S システム上で動作するサーバ側のプログラムおよびバッチシステムの開発に適しています。テンプレート、データ定義、プログラム定義を使い、C/S システムのサーバプログラムと、クライアントプログラムに必要なサーバプログラムとのリンクのインタフェースを効率良く作成できます。

SEWB+/OLTP DEFINER

C/S システムの構築に必要なクライアントアプリケーションとサーバアプリケーションのインタフェース設計およびアプリケーションの機能の配置設計を支援するツールです。

SEWB+/RECORD DEFINER

SEWB+ 基本開発環境のレコード設計支援機能のことです。

SEWB+/REPOSITORY の辞書情報を使用して、レコード情報を定義できます。

SEWB+/REPORT MANAGER

SEWB+ で定義した情報の印刷を支援するツールです。SEWB+/CONSTRUCTION と連携してプログラム処理概要図を、SEWB+/REPOSITORY と連携してインバクトレポートやデータ項目または業務ルールの定義内容などを印刷できます。これらのドキュメントを定義内容の確認や、保守作業に役立てられます。

SEWB+/REPOSITORY

SEWB+ 基本開発環境のリポジトリ管理機能のことです。

SEWB+/REPOSITORY は、リポジトリに格納された資源（ドキュメントと辞書）、およびその資源間の関連をサーバ上で統合管理します。また、SEWB+/REPOSITORY と SEWB+/REPOSITORY-BROWSER を合わせて SEWB+/REPOSITORY と呼ぶこともあります。

SEWB+/REPOSITORY-BROWSER

SEWB+ 基本開発環境のリポジトリブラウザ機能のことです。

SEWB+/REPOSITORY-BROWSER を使うと、リポジトリのシステム開発資源を、クライアントでビジュアルに操作できます。SEWB+/REPOSITORY-BROWSER では、リポジトリへ辞書やドキュメントを登録したり、目的の資源を検索したり、資源間の関連を手がかりにブラウジングしたりできます。

SEWB+ 基本開発環境

次に示す機能を一つに統合したツールです。

- リポジトリ管理機能（SEWB+/REPOSITORY）
- リポジトリブラウザ機能（SEWB+/REPOSITORY-BROWSER）
- プログラム構築支援機能（SEWB+/CONSTRUCTION）
- レコード設計支援機能（SEWB+/RECORD DEFINER）

SEWB+ 基本開発環境セット

SEWB+ 基本開発環境と Groupmax Object Server から構成される製品です。

SI 支援ユティリティ

システムバージョンの機能を利用して、異なるシステムバージョンに関連づけられたドキュメントのうち差分のあるドキュメントを、比較・抽出するためのユティリティです。

TP モニタ

トランザクション処理の監視、および制御をするソフトウェアのことです。オンラインシステムを構築するための基盤になる機能を提供しています。主な機能として、通信機能、スケジュール機能および障害発生時の回復機能があります。

UOC リバース機能

ソースファイルで直接編集された UOC（ユーザ追加処理）を、プログラム定義ファイルの UOC に取り込む機能です。

XMAP3

基幹業務で利用する画面や帳票を作成するためのツールです。XMAP3 では、画面や帳票の作成から、表示・印刷までを一貫して支援しています。なお、XMAP3 で開発した表示・印刷の AP は、Windows や UNIX のスタンドアロン環境のほかに、C/S システム環境やホストと連携した環境でも運用できます。

(ア行)

インポート

エクスポートで取り出された資源を、リポジトリに取り込むことをいいます。各サーバに分散された資源を収集する場合などに利用します。

エクスポート

リポジトリの資源を、ハードディスク上のファイル、フロッピーディスク、カートリッジ磁気テープなどの外部媒体に取り出すことをいいます。ルートドキュメントフォルダ単位、または辞書フォルダ単位でエクスポートできます。ほかのサーバに資源を配布する場合などに利用します。

エンティティ

ER 図で、データ項目などの実態を表すものです。

(カ行)

開発方式設計

システム処理方式設計で決定した内容を基に、C/S システム上の各 AP を開発するための開発方式を設計する作業です。この工程では、開発言語、支援ツール、SEWB+ で使用するテンプレートおよび部品の開発計画を決定します。

関連

データ項目とドキュメント、データ項目と業務ルールというように、資源間に付けられた依存関係をいいます。関連は、リポジトリブラウザを使用して任意に設定できる「ユーザ関連」と、SEWB+ ツールによって自動的に設定される「関連」に大別されます。リポジトリブラウザの関連ブラウザ機能を使用すると、関連づけられた資源をブラウジングでき、変更波及範囲の解析などに利用できます。

共通部品設計

パターンテンプレートおよび部品テンプレートを設計することです。システム処理方式で決定した機能分散の方式をプログラム単位に分割して、各プログラムの処理を明確にします。その中で、処理内容が類似するものや、複数の AP 間で利用されるロジックはパターンテンプレートや部品テンプレートとして抽出し作成します。このとき、既存のテンプレートや SEWB+ で提供している標準テンプレートを適用するかどうかを検討します。

業務機能設計

業務処理を実現するための、コンピュータ処理を構成する種々の機能について、その処理内容、入出力、信頼性および性能を明らかにすることです。

業務ルール

データ項目に着目して、データ項目特有の処理を部品化したものです。SEWB+/CONSTRUCTION を使ったアプリケーション開発で共通に利用することを目的としています。業務ルールは必ずデータ項目と関連づけて登録し、業務ルール辞書で管理します。実際にソースプログラムに展開する処理の内容は、ルールスクリプトに記述します。業務ルールに適した処理としては、データ項目の値や形式の妥当性をチェックする処理、形式を変換する処理、編集処理、複数のデータ項目で成り立つ計算処理などが挙げられます。

固定システムバージョン

「システムバージョン」を参照してください。

(サ行)

システム化機能詳細設計

サーバ AP 作成時に作成される「プログラム関連図」、「ファイル・レコード仕様書」および「各プログラムの処理概要図」の情報をまとめたものです。SEWB+/CONSTRUCTION を使用したプログラム構造設計の工程では、入力情報となります。

システム化処理内容設計

プログラム関連図に記述した各プログラムについて、入出力と処理の概要をまとめたものです。

システム基本設計

システムの外部仕様および内部仕様を決定する作業です。なお、このマニュアルではシステム基本設計をシステム設計としています。

システム詳細設計

システム基本設計で決定した内容を基に、プログラム作成で使用する辞書、ユーザインタフェース、ファイルおよび DBなどを設計します。

システム処理方式設計

業務機能設計で設定したシステム要件を基に、C/S システム上での機能の配置や処理の流れを設計します。

システムバージョン

システムを構成するドキュメント群を管理するためのバージョンです。管理目的に応じて種別（浮動/固定）を切り替えられます。

- 「浮動」は、常に最新のバージョンを示します。開発中のドキュメントなどに設定します。
- 「固定」は、設定された特定のバージョンを示します。開発を完了したドキュメントなどに設定します。

(夕行)

チェックアウト

リポジトリのドキュメントを、クライアントのファイルシステムに取り出すことです。

チェックイン

チェックアウトで取り出されたドキュメントを、リポジトリに格納（返却）することです。

データ制約

データ分析時に抽出される、データ項目に共通する処理のことです。SEWB+/REPOSITORY の辞書に業務ルールとして登録されます。制約データはデータ項目の分類によって「項目固有制約」と「項目関連制約」に分けられます。また、処理のタイプによって「チェック」と「導出」に分けられます。これらの分類によって、辞書に登録されときの登録方法が変わります。

データ層

DB と処理システム間の連絡を取る部分のシステム（AP の集まり）です。データ層を作成することで、DB 種別を意識しない運用が実現されます。

データ中心アプローチ

データを企業全体での重要な財産であると考え、共有資源として扱うシステム開発の方法論です。企業情報システムの基盤整備のために必要な方法論といえます。データ中心アプローチを採用したシステム開発では、先にデータを共有資源として分析して標準化したあと、システムやプログラムを設計、開発します。

データ定義

SEWB+/CONSTRUCTION での AP 作成に使用するファイルや DB を設計します。ファイルや DB の設計には、辞書のレコード定義（最上位結合項目）や SEWB+/RECORD DEFINER で作成したレコード定義を利用します。

データモデル

データをどのように整理し、関連づけて DB に格納するかというモデル化の手段を与えるものです。ユーザから見えるデータの論理的構造と、データについての操作方法を規定するものがあります。

テンプレート

パターンテンプレートともいいます。テンプレートは、生成時に AP の枠組みになるプログラム構成を記述したものです。部品や UOC の挿入場所、可変項目への値の設定の指示なども記述されています。テンプレートはテンプレート記述言語と使用するプログラミング言語で記述します。

ドキュメント

リポジトリで管理されるシステム分析・設計情報、およびプログラム開発情報などを保管するファイルです。ドキュメントの例としては、SEWB+ ツールで作成したシステム分析図やデータ定義情報、ソースプログラム、または Microsoft Word などで作成した企画文書、設計文書類があります。

ドキュメントフォルダ

リポジトリでドキュメントをまとめて管理するための入れ物です。ディレクトリに相当します。

(ナ行)

内部インタフェース設計

プログラム中で利用する中間ファイルなどに対し、ファイル編成やレコード形式などの属性情報を設定し、ファイルRPC インタフェースおよび内部テーブルなどの各仕様を作成することです。

(ハ行)

ビジネスフロー図（業務処理概要図）

業務機能設計で作成されます。新しく開発するシステムの構想を基に、システム全体の運用を表したものです。

開いた関連

エクスポート対象になるドキュメントとエクスポート対象外になるドキュメントの間に関連が付けられている場合、この関連を開いた関連といいます。これに対し、エクスポート対象のドキュメント間に付けられている関連を閉じた関連といいます。

ファンクション層

一つ、または複数の AP から構成される、システムの主処理部分です。プレゼンテーション層およびデータ層と連携して、業務を処理します。

浮動システムバージョン

「システムバージョン」を参照してください。

部品

部品テンプレートともいいます。複数の AP 間で利用されるロジックです。テンプレート記述言語と使用するプログラミング言語で記述します。部品は、ソースプログラム生成時に、テンプレートに取り込まれて展開されます。

プレゼンテーション層

ユーザインタフェース部分です。

プロトタイプ

プロトタイプには、外部仕様を確認するための業務プロトタイプと、SEWB+ でのシステム開発を評価するためのプロトタイプがあります。業務プロトタイプはユーザインタフェース基本設計で作成します。システム評価用のプロトタイプは、SEWB+ 導入の前提作業として、実現性、性能、生産性および手順の妥当性などを評価するために開発します。

(マ行)

マップ定義ファイル

XMAP3 で作成した画面や帳票の定義情報を格納したファイルのことです。SEWB+/CONSTRUCTION では、マップ定義ファイルを利用したクライアント AP の作成もできます。

(ヤ行)

ユーザインタフェース基本設計

外部仕様を早期に固めて、効率良く作業を進めるために必要な作業です。ここでは、画面遷移図や画面・帳票レイアウトを設計し、プロトタイプを利用してユーザとの意志確認をします。この作業での検討結果はユーザインタフェース詳細設計に反映されます。

ユーザインタフェース詳細設計

ユーザインタフェース基本設計の結果を受けて、画面・帳票を GUI 構築ツールや帳票作成ツールを利用して作成し、詳細なエラー処理や編集処理を作成します。

ユーザ追加処理

ユーザが作成するコーディングで、主にテンプレートや部品では対応できない AP 独自仕様の部分です。ユーザ追加処理はプログラム定義ウィンドウで編集します。テンプレート中にユーザ追加処理の挿入位置を記述しておくこと、プログラム定義ウィンドウ上に反映され、表示されます。ユーザ追加処理は UOC (User Own Coding) と表記されることもあります。

(ラ行)

リポジトリ

システム開発の各工程で発生する情報の保管庫です。このシステム開発資源 (ドキュメントと辞書) と、その資源間の関連をサーバ上で統合管理します。リポジトリによって、クライアントで動作するアプリケーションで、システム開発資源を共有したり連携したりできます。

リポジトリブラウザ (SEWB+/REPOSITORY-BROWSER)

SEWB+ 基本開発環境のリポジトリブラウザ機能のことです。

リポジトリブラウザは、リポジトリの資源をクライアントでビジュアルに操作できる機能です。リポジトリに格納された資源をツリー形式やリスト形式で表示できます。リポジトリブラウザでは、リポジトリにドキュメントや辞書を登録したり、目的の資源を検索したり、資源間の関連を手がかりにブラウジングしたりできます。

リレーションシップ

データモデルの構成要素の一つです。データ項目などの実態を表すエンティティ同士がどのような関連で結ばれているかを表します。エンティティとリレーションシップで表現されるデータモデルの代表的なものに ER 図があります。

レコード生成キーワード

データ定義で定義したレコード構造を COPY メンバファイルまたはヘッダファイルとして生成する場合に参照されるキーワードです。

レコード定義

レコードの構造を定義した最上位結合項目またはレコード定義ファイルのことです。レコード定義は、SEWB+/RECORD DEFINER を利用すれば、効率良く登録できます。レコード定義は、次のアプリケーション開発の場面で利用されます。

- SEWB+/CONSTRUCTION でのデータ定義
- EUR でのレポート設計

索引

A

at コマンド 138

C

C/S システム〔用語解説〕 143

C/S 処理フロー設計〔用語解説〕 143

C/S 処理フローの設定 19

COBOL2002/TD 76

COBOL 開発マネージャ 74

D

DB での環境設定 93

DB の初期化 93

DOA〔用語解説〕 143

DOA 手法 3

DOA 手法〔用語解説〕 143

E

ER 図〔用語解説〕 143

EUR 45, 62

EUR〔用語解説〕 143

O

Object Server〔用語解説〕 143

ODBC などを利用した C/S システム 9

R

RPC 60

RPM_{MaxU_{sr}} 138

S

SEWB+/CODE-DESIGN 13

SEWB+/CONSTRUCTION 13

SEWB+/CONSTRUCTION〔用語解説〕 143

SEWB+/OLTP DEFINER 13, 60

SEWB+/OLTP DEFINER〔用語解説〕 143

SEWB+/RECORD DEFINER 13

SEWB+/RECORD DEFINER〔用語解説〕 143

SEWB+/REPORT MANAGER 13

SEWB+/REPORT MANAGER〔用語解説〕 143

SEWB+/REPOSITORY 13

SEWB+/REPOSITORY〔用語解説〕 144

SEWB+/REPOSITORY-BROWSER 13

SEWB+/REPOSITORY-BROWSER〔用語解説〕 144

SEWB+/STANDARD-SUBROUTINE 14

SEWB+ COBOL 構造図エディタ 13

SEWB+ 基本開発環境 4, 13

SEWB+ 基本開発環境〔用語解説〕 144

SEWB+ 基本開発環境セット 4, 13

SEWB+ 基本開発環境セット〔用語解説〕 144

SI 支援ユティリティ〔用語解説〕 144

SI 支援ユティリティとは 81

SI 支援ユティリティの利用 81

T

TP モニタ〔用語解説〕 144

TP モニタを利用した C/S システム 8

U

UOC リバース機能〔用語解説〕 144

V

Visual Basic 42, 44

W

Web システム 10

Windows Server での環境設定 93

X

XMAP3 42, 43

XMAP3〔用語解説〕 144

あ

アクセス権 90

い

インタフェース仕様 61

インポート〔用語解説〕 144

う

運用環境の構築 94

え

エクスポート〔用語解説〕 144

エクスポート・インポートの自動運用 138

エンティティ〔用語解説〕 144

お

オリジナル配布 115

か

開発環境の構築 22
開発言語および支援ツールの決定 21
開発方式設計〔用語解説〕 145
開発方式の設計 22
開発ライフサイクル 134
各種基準書の設計 21
カスタマイズ 32
カスタマイズ仕様および新規テンプレート仕様の検討
51
環境の初期化 93
関連 89
関連〔用語解説〕 145

き

基幹システム 2
机上検証 20
基盤ソフトウェア構成・システム処理方式の確定 20
共通部品開発計画の設定 22
共通部品設計〔用語解説〕 145
業務機能設計 19
業務機能設計〔用語解説〕 145
業務処理概要図 19
業務ルール〔用語解説〕 145
業務ルール辞書 3, 35

く

組み合わせテスト方式設定 22
グループ登録ファイルの設定 94

け

継承関係を利用した登録 36
継承項目 33
結合項目（レコード項目） 33

こ

更新履歴での管理 88
項目関連制約 24, 35
項目共通制約 35
項目固有制約 24, 35
固定システムバージョン〔用語解説〕 145

さ

最大ログインユーザ数変更コマンド 138
参照専用チェックアウト 89

し

辞書 2
辞書フォルダ 88
システム化機能詳細設計〔用語解説〕 145
システム化処理内容設計〔用語解説〕 145
システム基本設計〔用語解説〕 145
システム共通定義ファイル 93
システム詳細設計〔用語解説〕 145
システム処理方式 19
システム処理方式設計〔用語解説〕 145
システム設計の概要 18
システムで共用する業務ルールの登録 37
システム統合 134
システムバージョン〔用語解説〕 146
システムバージョンでの管理 88
システムバージョンの利用 80
実機テスト 134
修正時の手順 69
初期設定パラメタファイル 93
処理パターンの抽出 51

せ

制約 23
絶対パスでの設定 64

そ

相対パスでの設定 64

た

代表シナリオ 26
単項目（基本項目） 32

ち

チェック 24, 35
チェックアウト〔用語解説〕 146
チェックイン〔用語解説〕 146
帳票出力する AP の設計 62

て

データ項目辞書 2
データ制約〔用語解説〕 146
データ制約定義の確認 24

データ制約の分類ごとの設定 24
 データ層 8, 9, 10
 データ層〔用語解説〕146
 データ中心アプローチ〔用語解説〕146
 データ定義〔用語解説〕146
 データモデリング 28
 データモデル 28
 データモデル〔用語解説〕146
 テーブル仕様 61
 適用条件 36
 テストデータの作成 26
 テンプレート 3, 51
 テンプレート〔用語解説〕146
 テンプレート適用の検討 60

と

統合テスト 80
 導出 24, 35
 ドキュメント〔用語解説〕146
 ドキュメント種別の登録 93
 ドキュメントフォルダ 88
 ドキュメントフォルダ〔用語解説〕146
 ドキュメントフォルダ構成 80
 特定業務で扱う業務ルールの登録 37
 ドメイン制約 35

な

内部インタフェース設計〔用語解説〕147

ね

ネットワーク機能 89
 ネットワーク機能利用のアクセス権 92

は

バージョン 102
 パス設定 64
 バッチシステム 10
 版 102

ひ

ビジネスフロー図(業務処理概要図)〔用語解説〕147
 ビジネスルール 23
 評価項目の設定 26
 標準テンプレート適用の検討 51
 開いた関連 121
 開いた関連〔用語解説〕147

ビルド 76

ふ

ファイル関連の登録 74
 ファイル仕様 61
 ファンクション層 8, 9, 10
 ファンクション層〔用語解説〕147
 複数パスの設定 65
 複製配布 115
 浮動システムバージョン〔用語解説〕147
 部品 3, 51
 部品〔用語解説〕147
 プレゼンテーション層 8
 プレゼンテーション層〔用語解説〕147
 プログラム関連図の作成 60
 プロジェクトマスタファイルの登録 76
 プロトタイプ〔用語解説〕147
 プロトタイプでの検証 20
 分散化設計の確認 19

ま

マップ定義ファイル〔用語解説〕147

ゆ

ユーザインタフェース基本設計〔用語解説〕147
 ユーザインタフェース詳細設計〔用語解説〕147
 ユーザ関連 89
 ユーザ追加処理 51
 ユーザ追加処理〔用語解説〕148
 ユーザ登録 94
 ユーザ登録ファイルの設定 94

り

リビルド 76
 リポジトリ 3
 リポジトリ〔用語解説〕148
 リポジトリのアクセス権 90
 リポジトリの初期化 93
 リポジトリブラウザ(SEWB+/REPOSITORY-BROWSER)〔用語解説〕148
 リレーションシップ〔用語解説〕148

る

ルートドキュメントフォルダ 88
 ルールスクリプト 36

れ

- レコード形式の確認 61
- レコード生成キーワード〔用語解説〕 148
- レコード定義 46
- レコード定義〔用語解説〕 148
- レプリカ配布 114

わ

- ワークフローシステム 11