

アナリシスパターン勉強会 責任関係 事例紹介

株式会社 オーエスケイ
小井土 亨
(CBOP COM分科会 主査)

2000/07/19

Agenda

- システム開発概要
- 事例説明
 - システム要件(画面イメージ)
 - 組織型データ管理フレームワーク詳細
 - 人事情報管理システム詳細
 - フレームワーク利用カタログ

•略語説明

- FW フレームワーク
- CS カスタマイズシステム(実行可能な具体システム)
- IF インタフェイス(型)
- HIF ホットスポットインターフェイス
- CP コンポーネント

システム開発概要

背景「人事情報管理の要求」

- 人事システムの変化
 - 人事業務処理の支援から、戦略的人事管理へ変化してきている
- 複数の組織構造管理
 - 組織構造以外にプロジェクトなど複数の階層構造を管理できることを要求されている
- 組織構造の履歴管理
 - 組織構造の履歴管理ができるように要求されている

システム開発概要

目的「FWの有効性を実証する」

- FWによるシステム開発
 - 再利用性やカスタマイズ性の検証を行う
 - コンポーネントベース開発による、開発の効率化について検証を行う
- 利用ユーザごとのカスタマイズ
 - 管理するデータ項目は、自由に設定可能とする
 - 組織構造の履歴管理は意識する必要なし

システム開発概要

開発フェーズ

- 4段階のフェーズでシステムの開発を行う
 - 分析フェーズ
 - 設計フェーズ
 - インターフェイス設計フェーズ
 - コンポーネント設計フェーズ
 - 実装フェーズ
 - 実装モデル
 - インターフェイス実装
 - コンポーネント実装
 - 検査フェーズ

分析フェーズ概要

- 分析モデルの開発を行う
 - 分析モデルは、システム論理的な側面をモデル化する
- アナリシスパターンの応用
 - アナリシスパターンを応用することで、柔軟性と再利用性のあるモデルを作成する
 - アナリシスパターンは、モデル安定化の指針となる
 - 明解なモデルを探すための道しるべとして、アナリシスパターンを利用する

インターフェイス設計フェーズ概要

- インターフェイス設計フェーズ
 - インターフェイスモデル開発は、分析モデルの論理性を崩さずに実装可能なものに変換する
- システム基本構造
 - FWとCSの構造を決定する
 - 各FWの責任領域を明確にする
- インターフェイスの種類
 - FW・インターフェイス
 - ホットスポット・インターフェイス
 - CS用ローカルインターフェイス

人事情報管理システム概要

- 1998年6月
 - CBOPのプロジェクトとしてスタート
 - CBOP (Consortium for Business Object Promotion)
 - <http://www.cbop.gr.jp/>
- 1998年11月
 - 人事情報管理フレームワークの分析モデル作成完了
- 1998年11月～1999年1月
 - 人事情報管理フレームワーク内の階層型データ管理部分をフレームワークとして実装検証

人事情報管理システム概要

システム要件 その1

- 時系列組織構造の管理が可能
 - 組織構造(組織及び属している社員)を時系列で管理することができる
 - 任意に指定した時点の構造図を階層形式で表示することができる
- 組織図以外の階層管理が可能
 - 通常の組織構造以外に、プロジェクトの管理ができる

人事情報管理システム概要

システム要件 その2

- 社員の履歴情報の管理が可能
 - 所属履歴と職歴を管理することができる
- ドラッグ & ドロップによる所属変更
 - ドラッグ & ドロップ操作で、新しい所属の指定を行うことができる

人事情報管理システム概要

画面イメージ その1

- 99年1月28日時点の組織構造

人事情報管理システム

ファイル(F) 編集(E) 表示(V) 業務(B) ヘルプ(H)

組織構造 99/01/28

名前	よみがな	社員番号	誕生
柿崎 征典	カキザキ マサル	006666	1962
潮屋 光昭	シオヤ ミツアキ	007777	1965
山口 芳男	ヤマグチ ヨシオ	005555	1959
小井土 敦	コイト アツシ	004444	1961
富永 亨	トミナガ トオル	003333	1956

ヘルプを表示するには [F1] を押してください。

2000/07/19

人事情報管理システム概要

画面イメージ その2

- 97年1月1日時点の組織構造

人事情報管理システム

ファイル(F) 編集(E) 表示(V) 業務(B) ヘルプ(H)

組織構造 97/01/01

オーエスケイ

- 開発センター
 - 開発一課
 - 開発四課
- 技術支援センター
- CAC
 - 一課
 - 二課

名前	よみがな	社員番号	誕生
潮屋 光昭	シオヤ ミツアキ	007777	1965
山口 芳男	ヤマグチ ヨシオ	005555	1959
小井土 敦	コイト アツシ	004444	1961
富永 亨	トミナガ トオル	003333	1956

ヘルプを表示するには [F1] を押してください。

2000/07/19

人事情報管理システム概要

画面イメージ その3

- 社員情報登録画面

The screenshot displays two overlapping windows from a Japanese HR management system. The background window is titled '社員情報' (Employee Information) and has three tabs: '個人情報' (Personal Information), '所属履歴' (Affiliation History), and '採用情報' (Employment Information). The '個人情報' tab is active, showing the following fields:

- 社員コード: 004444
- フリガナ: コイト アツシ
- 氏名: 小井土 敦
- 生年月日: 61/11/11
- 性別: 0 - 男
- 本籍地: 10 - 群馬県

The foreground window is also titled '社員情報' and has the same three tabs. The '採用情報' tab is active, showing:

- 採用日: 84/04/01
- 採用区分: 0 - 新卒採用

Below these fields is a table with the following data:

NO.	入社日付	退社日付	勤務先名称
1	84/04/01		大塚商会
End			

Buttons for '追加' (Add) and '削除' (Delete) are located to the right of the table. At the bottom of the foreground window are 'OK' and 'キャンセル' (Cancel) buttons.

2000/07/19

人事情報管理システム概要

開発言語

- 階層型データ管理FW
 - Visual C++ (ATL) DLL形式
- 階層データ表示FW
 - Visual C++ (MFC) EXE形式
- 人事情報管理システム
 - UI Visual Basic DLL形式
 - データベース OODB C++

分析フェーズ

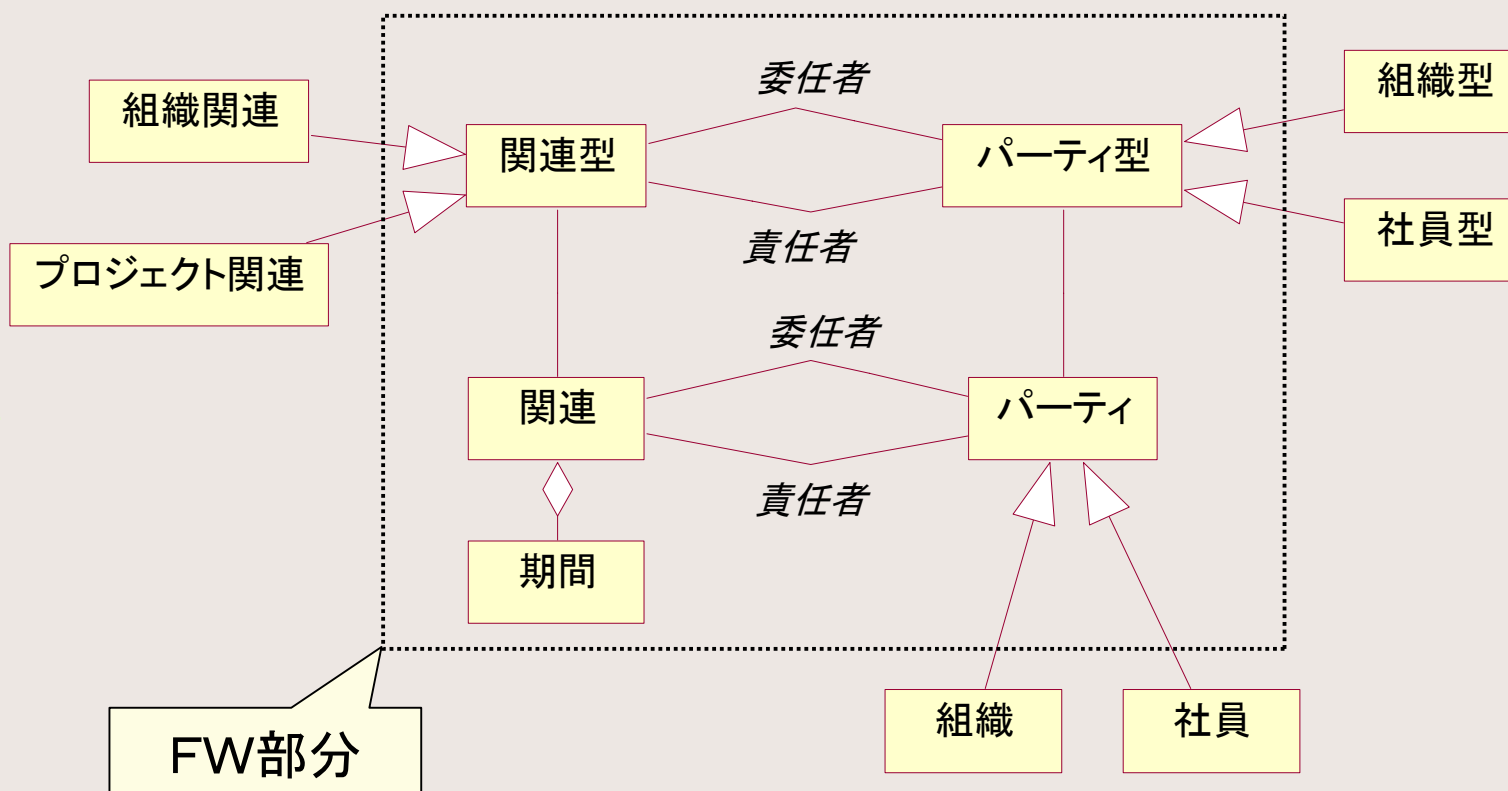
分析モデルの一般要件

- システムのメンタルモデルを作成する
 - 論理的な側面を明確にするモデル
- インターフェイス構造を明確にする
 - 分析モデルは、システムの実装ではなく、インターフェイスの指針となる
 - システムの論理側面と実装がつながることで、シームレスな開発や後戻り開発が可能となる
- 分析モデルのシステム開発への影響
 - モデルがシステムの柔軟性や再利用性に影響する
 - 特に、インターフェイスの安定は、フレームワークの適応範囲や再利用性の重要要件である

分析フェーズ 分析モデル要件

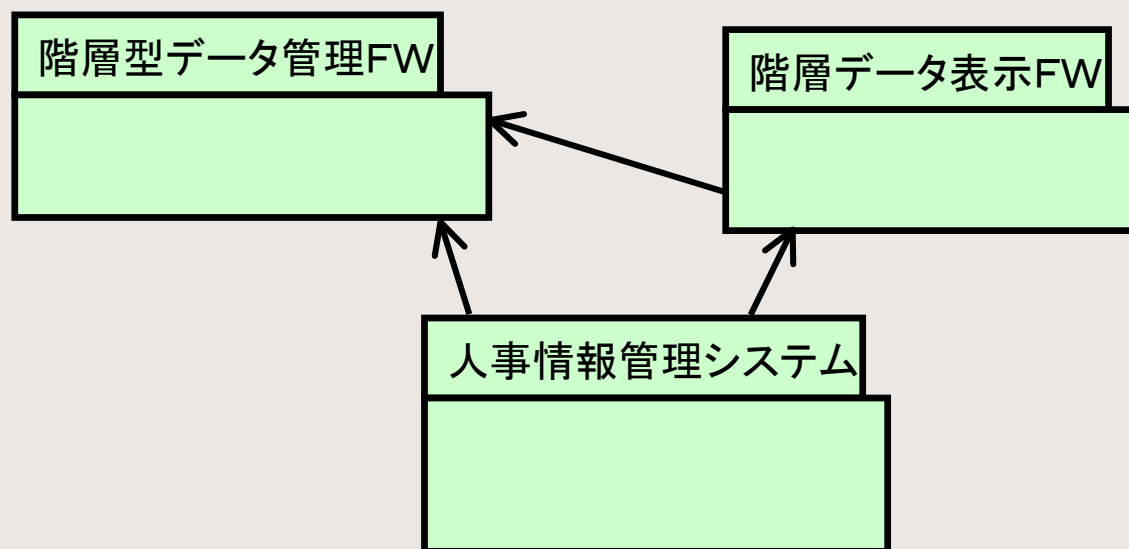
- モデルの柔軟性
 - モデルの基本構造を変えずに、階層構造の種類やデータの種類を変更できる
 - 上記の変更が発生しても、階層構造や階層構造履歴の管理部分に変更されない
 - 階層的な構造をもつシステムであれば、人事管理以外でも再利用可能とする
- インターフェイスの指針
 - FWのIFに、変換可能なモデルである
 - FWで想定するカスタマイズ可能部分(HIF)を明確にする

分析フェーズ 分析モデル



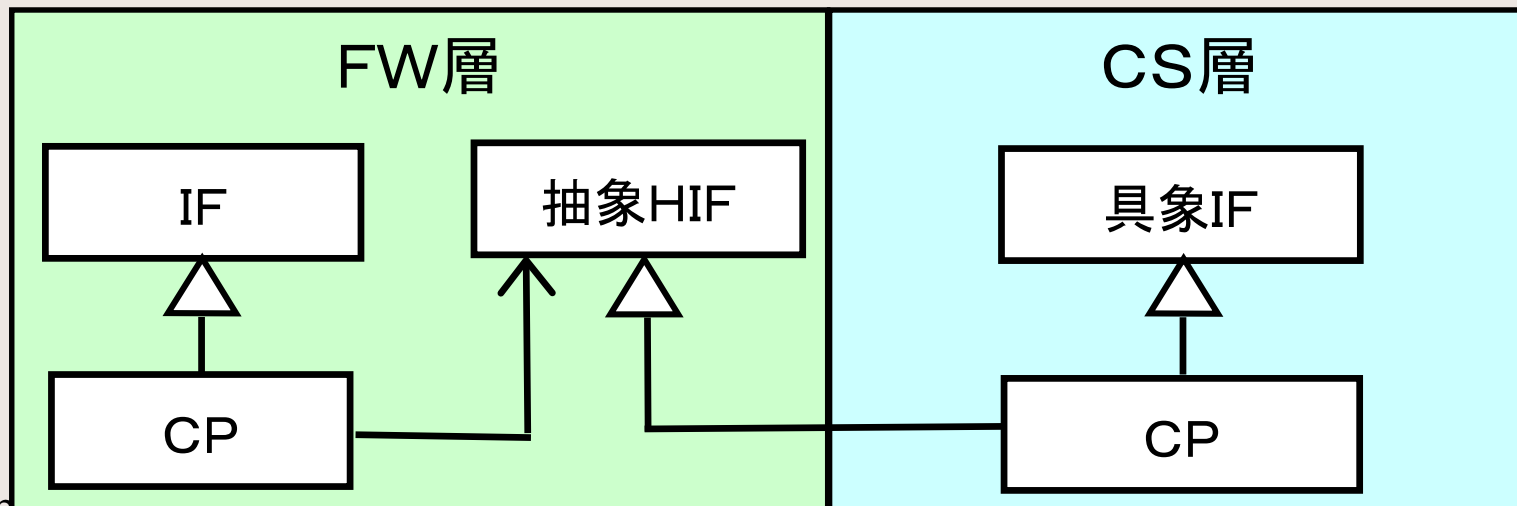
設計フェーズ システム構成

- システム構成を定義する
 - FWの種類や構成を定義する
 - FWの役割範囲を明確にする



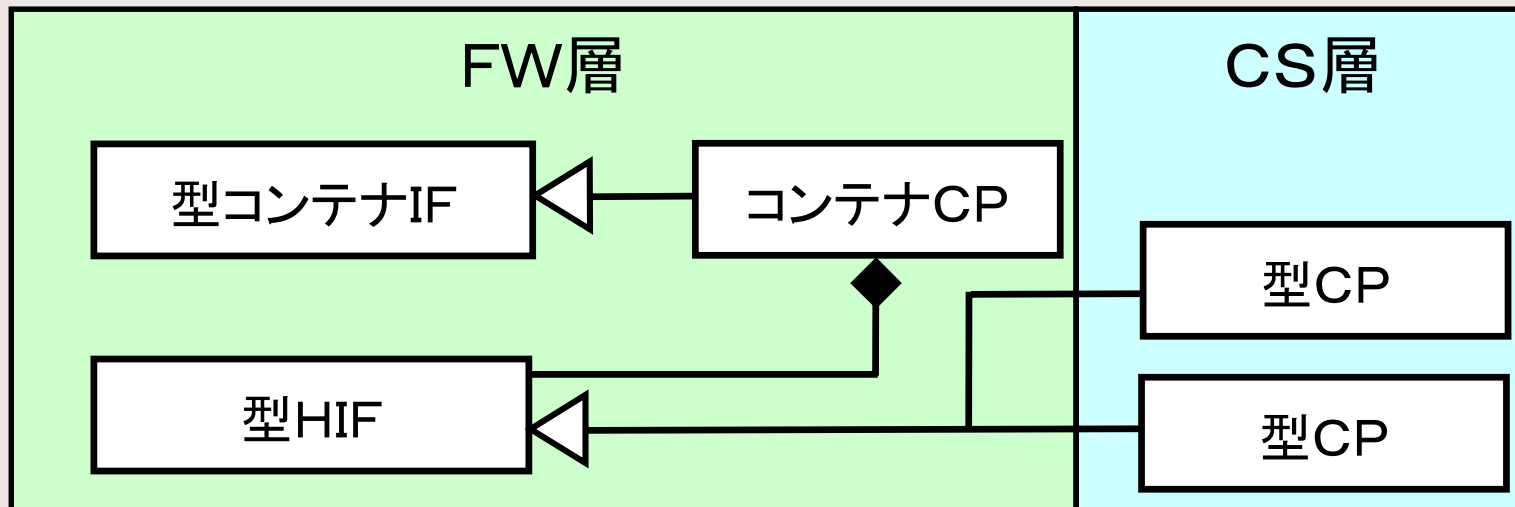
インターフェイス設計フェーズ 操作レベルオブジェクト

- 操作レベルオブジェクトの振分
 - FWとCSへ振分を行う
- CSへ振り分ける場合
 - 操作レベルのオブジェクトは、インターフェイス多重継承を利用して以下のような構造にする



インターフェイス設計フェーズ 知識レベルオブジェクト

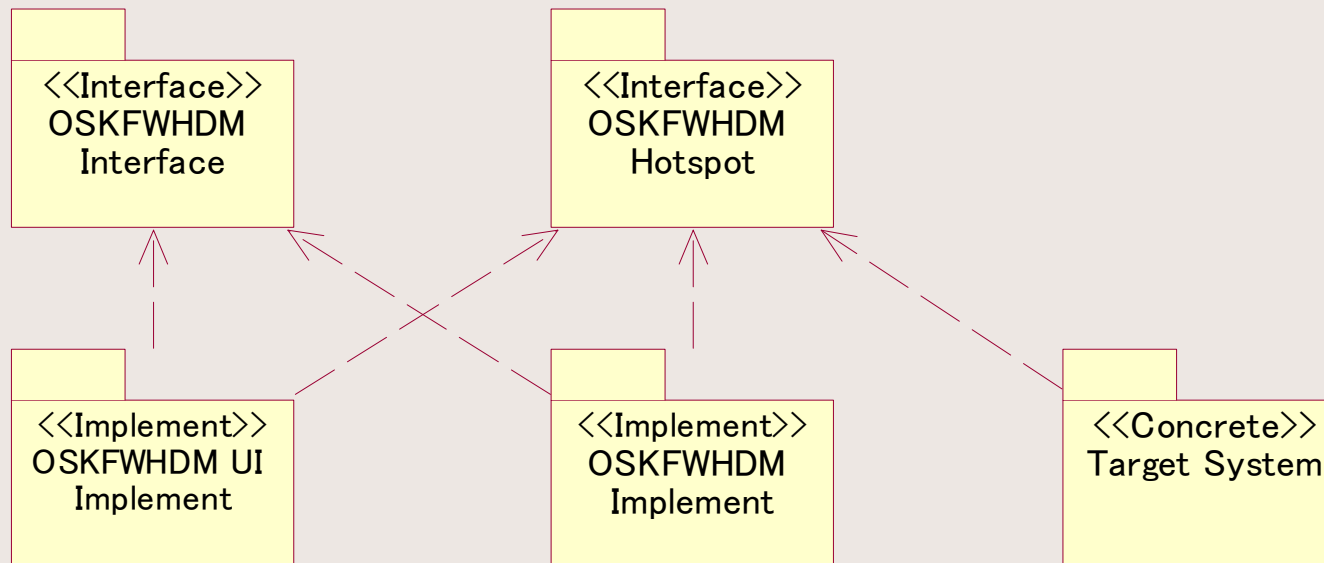
- 知識レベルオブジェクトの振分
 - 基本的にはCSへ振分を行う
- CSへ振り分ける場合
 - CSに振り分けられる知識レベルオブジェクトは、以下のような構造にする



組織型データ管理FW詳細 特長

- 組織型データの管理
 - 階層型データエンティティの整合性を維持する機能を提供する
- 複数の関連型及びパーティ型に対応
 - 複数の関連型やパーティ型を使用できる
 - 動的に、関連型やパーティ型を扱うことが可能
- 永続データの制御
 - 作成、削除、修正の制御を行う

組織型データ管理FW詳細 パッケージ構成



組織型データ管理FW詳細 インターフェース一覧

カスタマイズシステム用

組織型データ管理
IHierarchyDataManager

関連フレームワーク用

組織型データ管理拡張版
IHierarchyDataManagerEx

関連型のコンテナ
IResponsibleRelationTypeContainer

パーティ型のコンテナ
IPartyTypeContainer

組織型データ管理FW詳細 ホットスポット一覧

カスタマイズシステム
ISystem

関連の型
IResponsibleRelationType

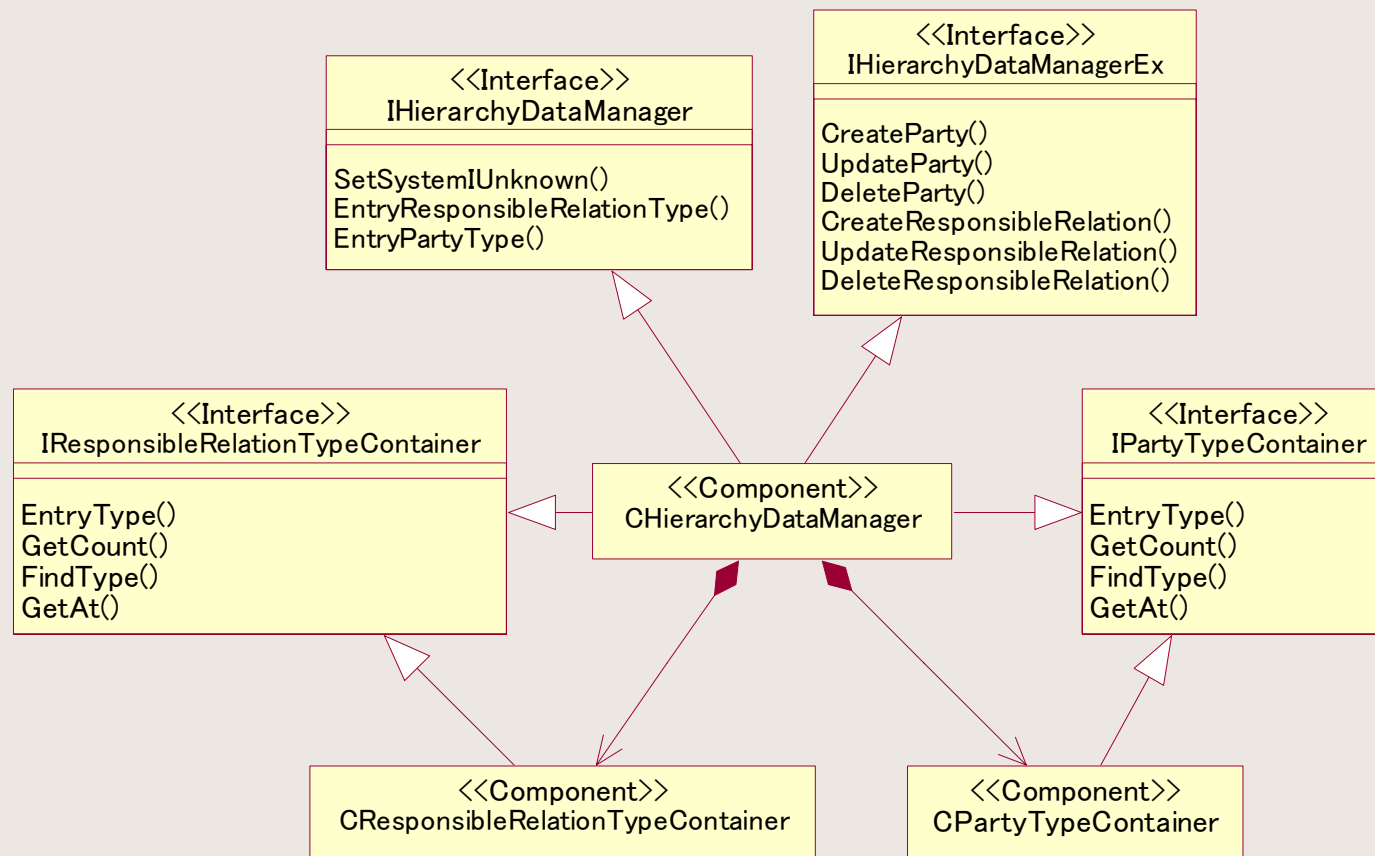
パーティの型
IPartyType

関連
IResponsibleRelation

パーティ
IParty

パーティのUI制御
IPartyProducer

組織型データ管理FW詳細 インターフェース-コンポーネント関連図



人事情報管理システム詳細 機能

- 組織情報の管理
 - 組織情報を管理する
 - 組織階層情報を管理する
 - プロジェクト階層情報を管理する
- 社員情報の管理
 - 社員情報を管理する
 - 社員の所属履歴を管理する
 - 社員の職歴を管理する

人事情報管理システム詳細 インターフェース一覧

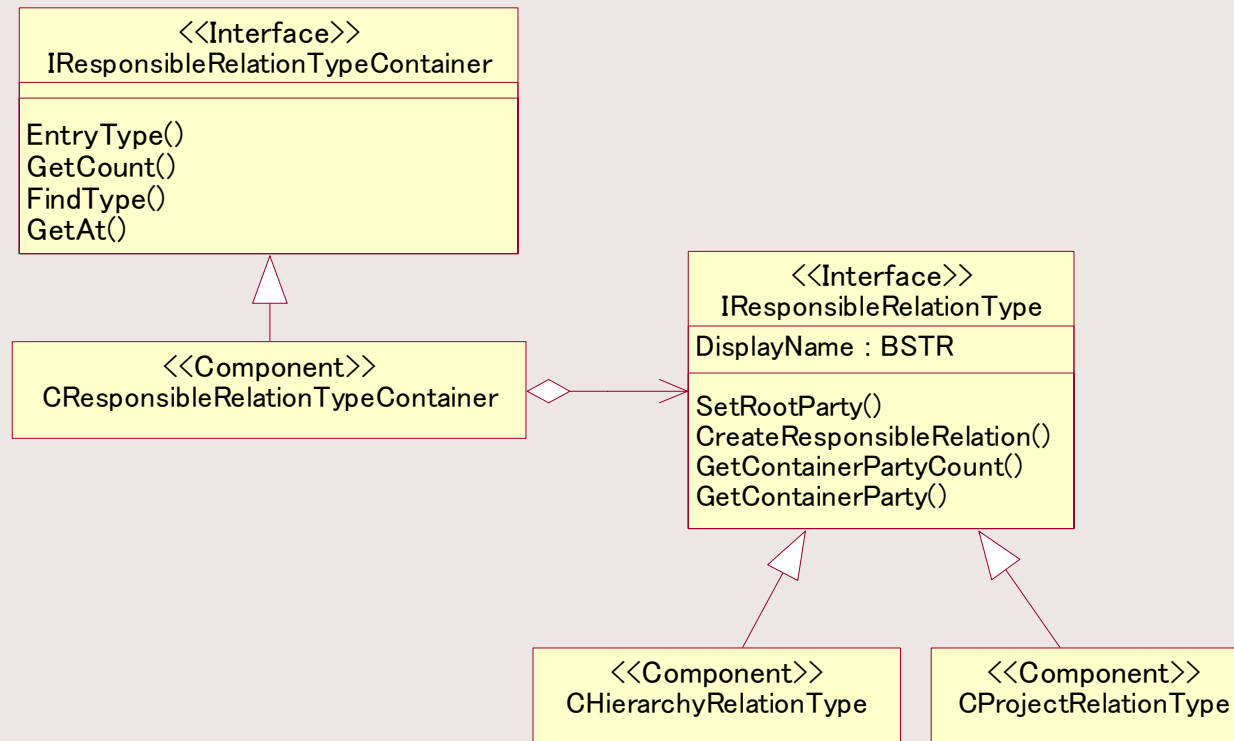
- カスタマイズシステム内部のローカルインターフェイス

組織
IOrganization

社員
IEmployee

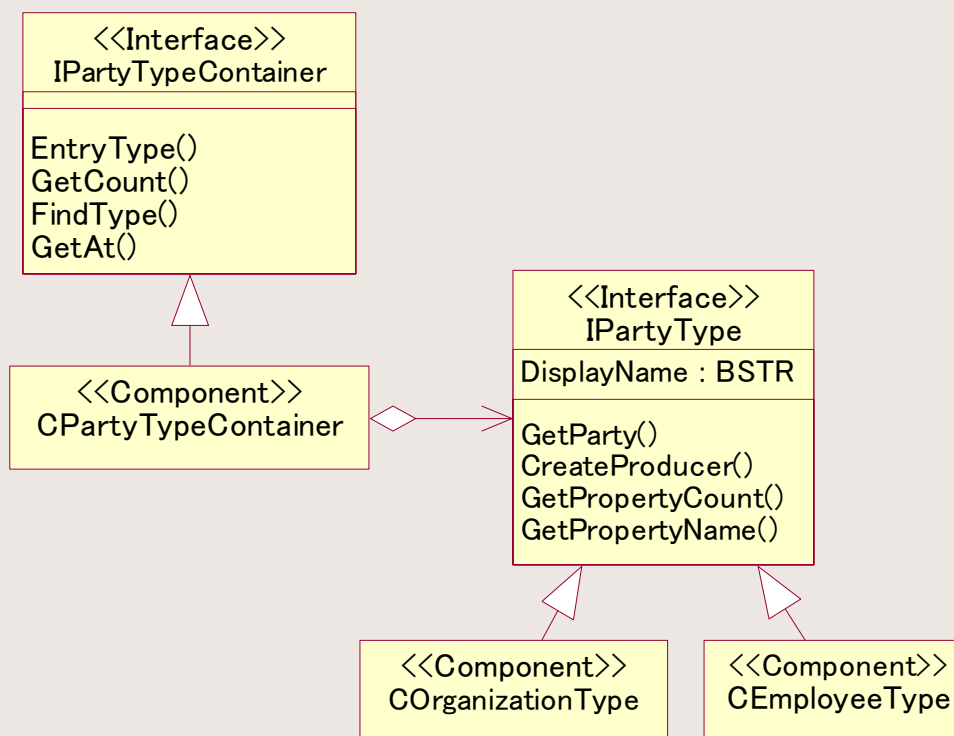
人事情報管理システム詳細 インターフェースーコンポーネント関連図

- 関連の関連図



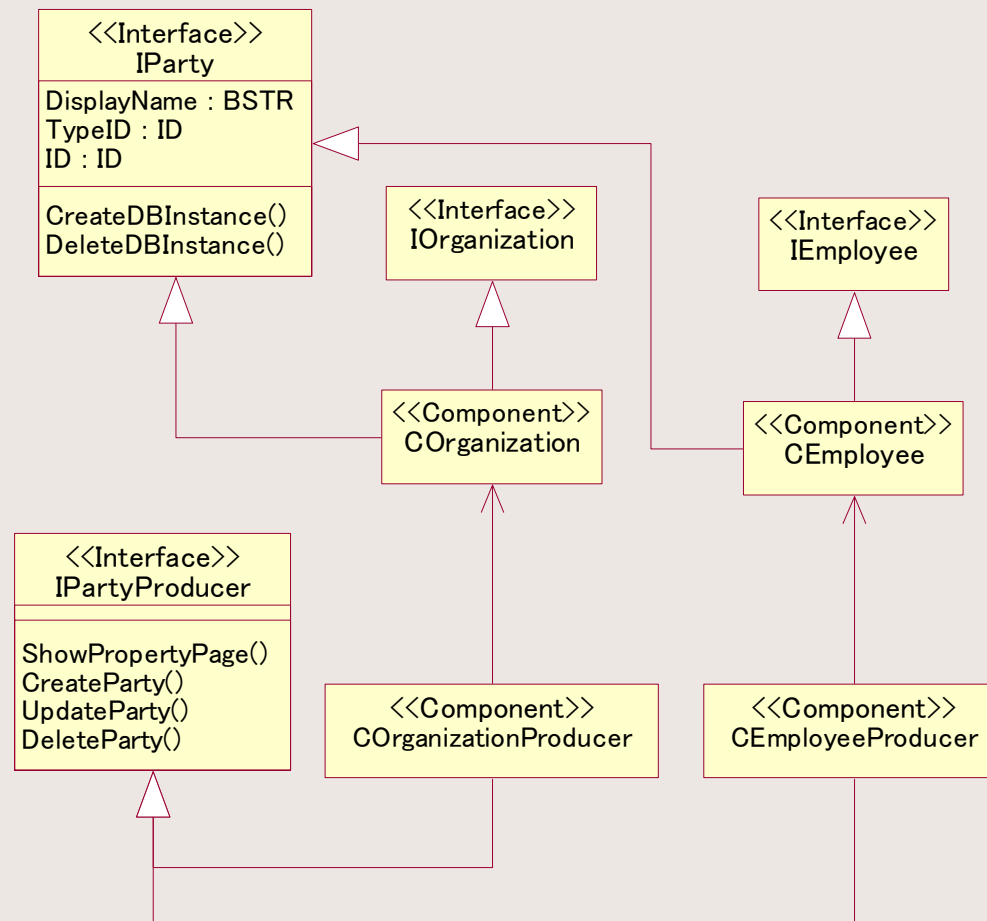
人事情報管理システム詳細 インターフェースーコンポーネント関連図

- パーティ型の関連図



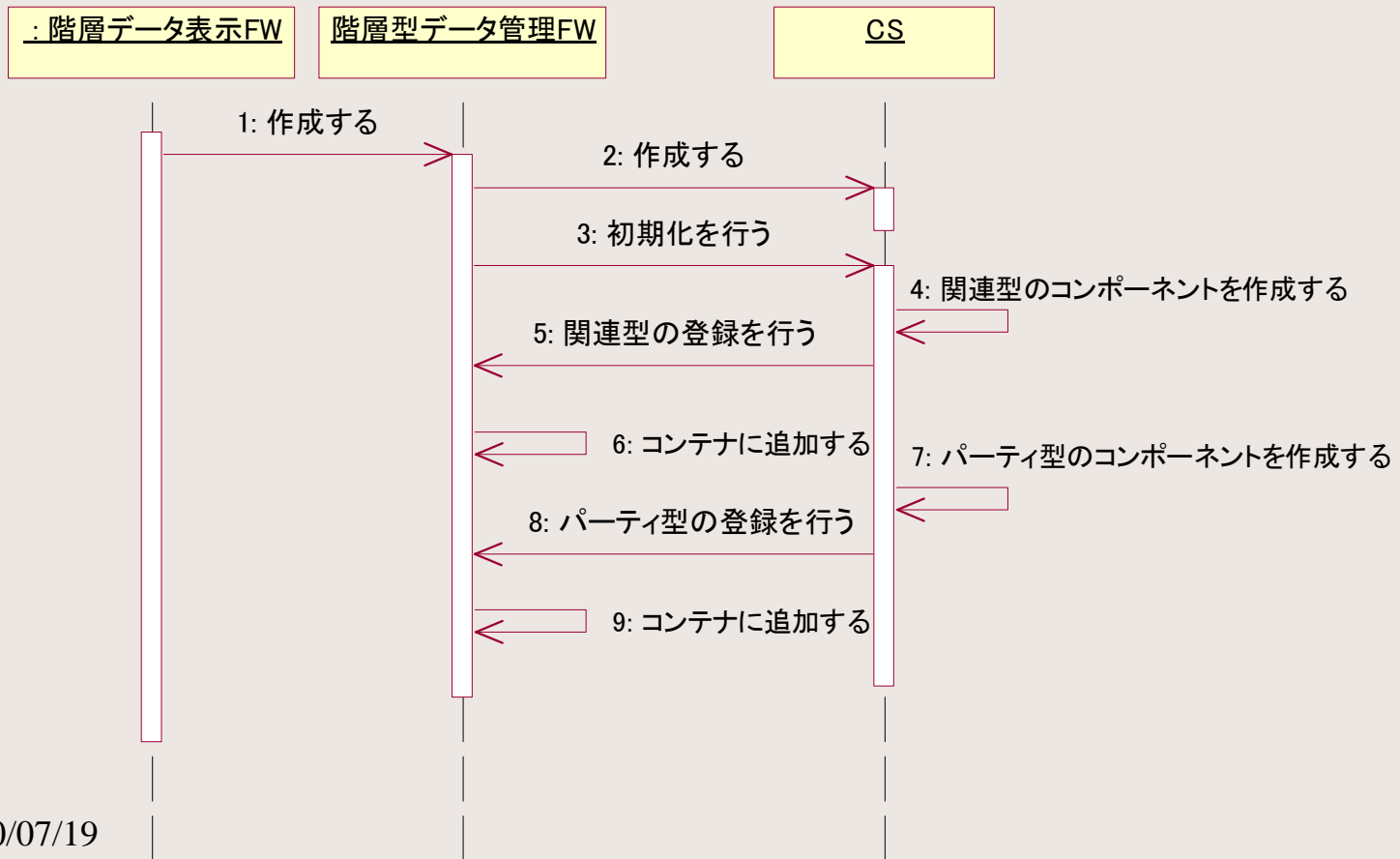
人事情報管理システム詳細 インターフェイス-コンポーネント関連図

- パーティの関連図



人事情報管理システム概要 シーケンス図

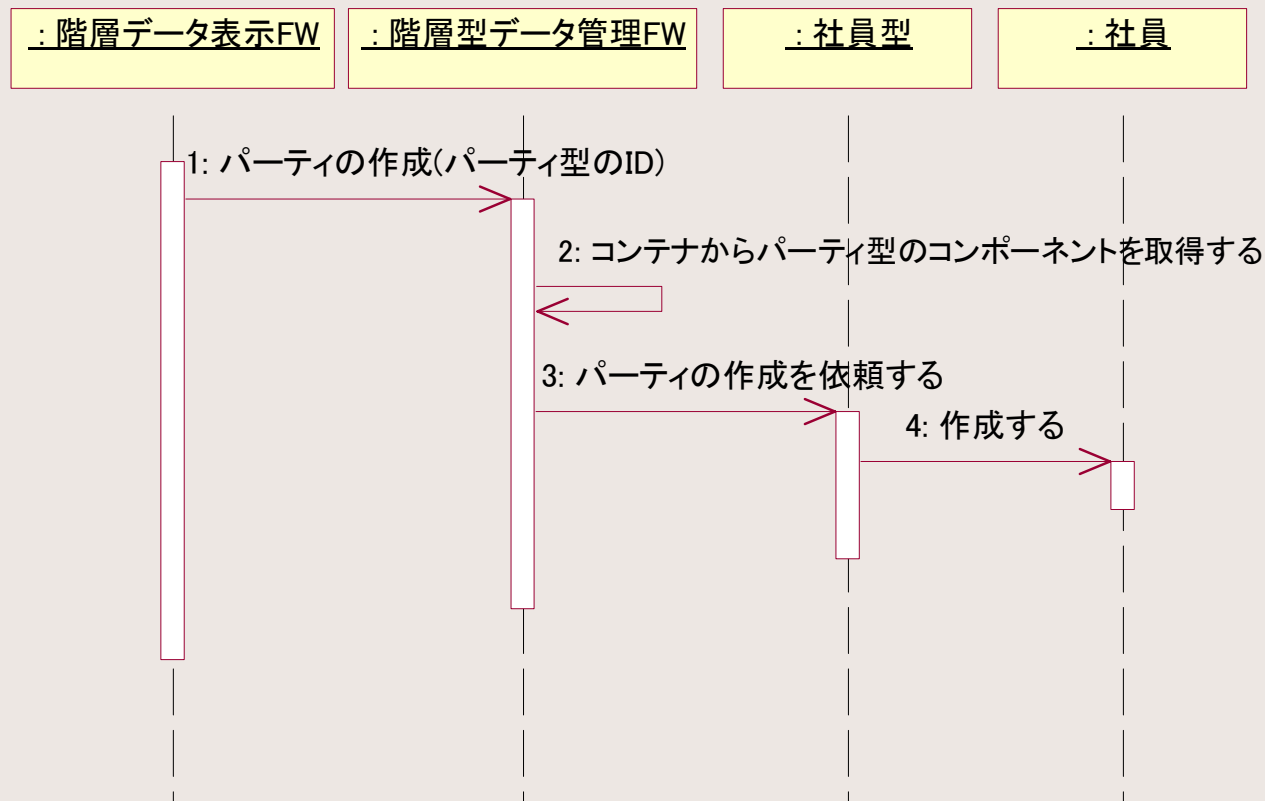
- 初期化処理



人事情報管理システム概要

シーケンス図

- 社員パーティの作成処理



人事情報管理システム概要

シーケンス図

- プロジェクト関連の作成処理

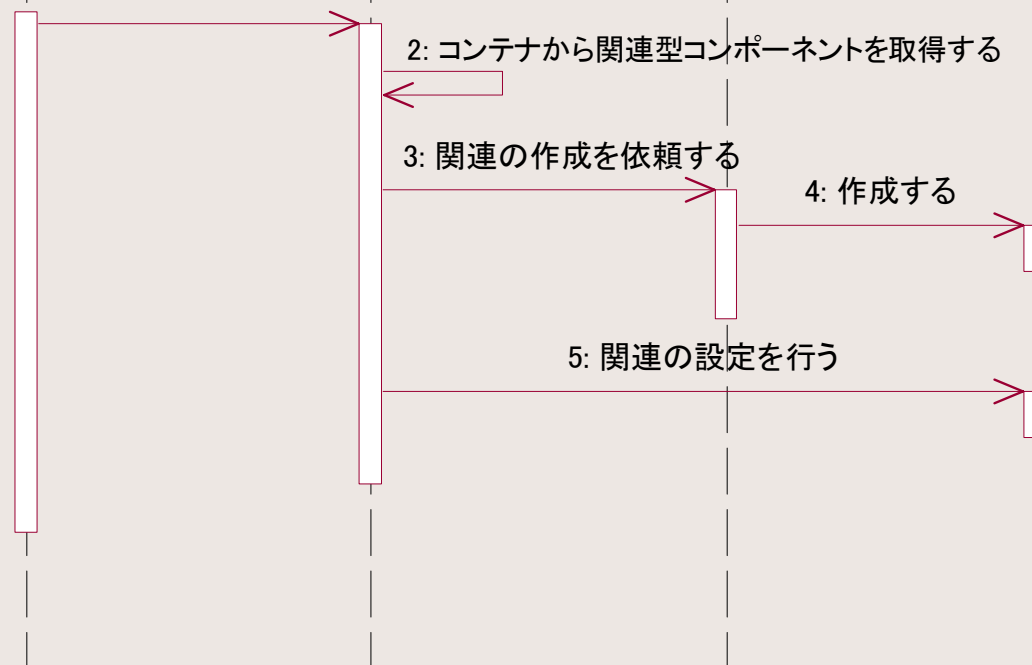
:階層データ表示FW

:階層型データ管理FW

プロジェクト関連型CP

:関連

1: 関連を作成する(関連型のID,責任者のパーティ型ID, 責任者のパーティID,委任者のパーティ型ID, 委任者のパーティID,)



2000/07/19

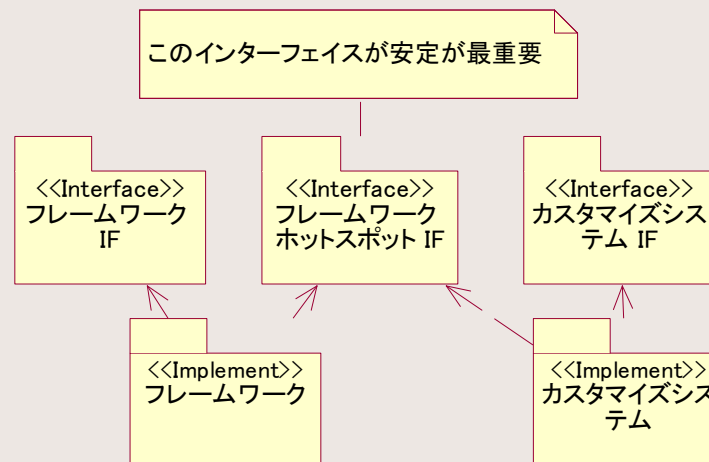
33

アナリシスパターン利用結果 責任関係パターン適応の利点

- パーティ間の関係が柔軟になった
 - 複数のパーティ間の関係が扱える
 - パーティ間の関係が増えても、FWの変更は不要
- パーティが柔軟になった
 - FWを変更することなく、パーティの種類が追加、変更可能
 - パーティ型のインスタンスにより、パーティにさまざまな制約を実装可能

アナリシスパターン利用結果 フレームワークの安定

- 分析モデルが安定する
 - アナリシスパターンはモデルの安定を主眼としている
- インターフェイスの安定
 - インファーフフェイスは分析モデルを実装したもの
 - 分析モデルの安定 = インターフェイスの安定
 - インターフェイスの安定により、FWとCSが個々に機能変更可能となる。



フレームワーク利用カタログ

階層型データ管理フレームワーク

- 目的
 - 階層型のデータを管理する機能を提供するフレームワークである。また、階層構造の履歴を管理する機能を提供する。
- 理由
 - 階層型のデータを管理する
 - 本フレームワークを利用すれば、複数の異なった種類の階層を持つ機能や構造を動的に変更する機能を実現することが可能になる。
 - 時系列のデータを管理する
 - 本フレームワークを利用すれば、履歴を一つの関係として管理するため、柔軟な履歴管理を実現することができる。また、関係が期間を持っているため、任意の時点の階層型のデータを参照することが可能となる。

フレームワーク利用カタログ

階層型データ管理フレームワーク(つづき)

- 適用可能性
 - 階層構造が複数ある場合
 - 動的に階層構造の種類を変更したり追加する場合
 - データを時系列の履歴で管理する場合
 - 階層構造の時間による変化を記録する場合
- 構造
 - パーティと責任関係
 - 格納するデータと格納されるデータに共通のパーティという抽象クラスを継承させることで、データを意識しないで自由に階層構造を構築できるようになっている
 - 責任関係に責任関係型を持たせることで、階層構造では階層の種類を意識せずに関係を構築できるようになっている

フレームワーク利用カタログ

階層型データ管理フレームワーク(つづき)

- カスタマイズ想定部分
 - データ種類
 - データの種類は、任意に定義することができる
 - データの処理
 - データの作成、表示及び修正処理はシステムが提供する
 - 階層関係の種類
 - 階層関係の種類は、任意に定義することができる
- 結果
 - 階層型で管理するデータ種類の拡張が容易である
 - 階層の種類が拡張が容易である
 - 階層構造の構築と維持が容易である