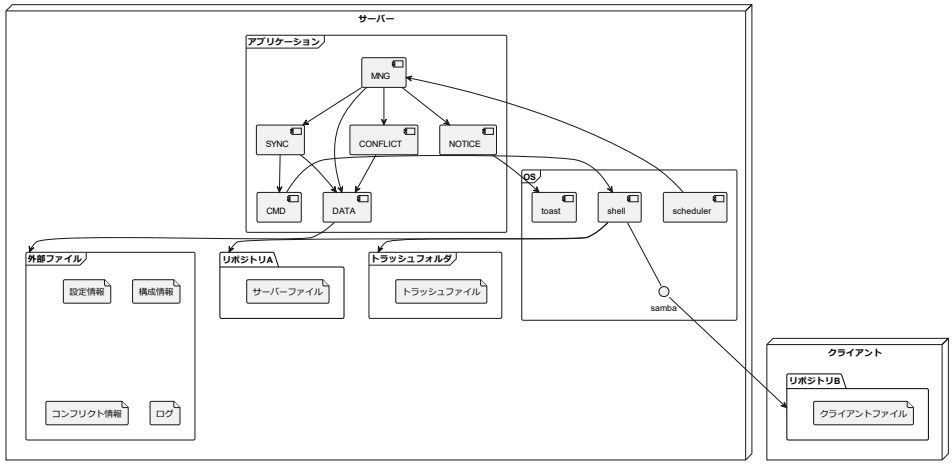
1.システム概要

1.1 実行環境

以下に実行環境を示す。

項目	プロダクト名	説明・制約など
サーバー	サーバーマシン	下記プロダクトを実行可能で、ユーザーからのアクセスに対し十分な処理能力を有するもの。 WindowsOS
	ファイルシステム	1.2TBの利用可能な容量があること。shellコマンドからのファイル・フォルダの追加・ 削除を許可する。
	Python	Python 3の実行環境があること
	shell	次の処理をアプリケーションから呼び出すことができること。・ファイルの削除・ファイルのコピー・ ファイルの追加・フォルダの作成・フォルダのリネーム・フォルダの削除・ファイルパスの取得
	toast	アプリケーションからOSの通知を呼び出すことができるもの
クライアント	クライアントマシン	下記プロダクトを実行可能で、ユーザーからのアクセスに対し十分な処理能力を有するもの。
	ファイルシステム	1TBの利用可能な容量があること
	samba	リポジトリを共有してサーバーから操作できる状態であること
通信	LAN	1Mbpsのトランザクションを処理できること

2.システム構成図



2.1 ユースケース検討

ファイルへの操作を定義し、その検出方法及びその場合に期待される処理を検討する

2.1.1 ファイルへの操作

ファイルに対する操作は以下の4種類に区分される。
ファイルに対するそれぞれの操作はソフト内でリポジトリのファイルと構成情報を比較することで検知する。

操作	概要	検出方法
操作なし	ファイルに対して何も操作しない	リポジトリのファイルパス、キャッシュの値が構成情報と合致する
追加	新たにファイルを追加する	構成情報に存在しないパスのファイルがリポジトリ上に存在する
削除	既存のファイルを削除する	構成情報に存在するパスのファイルがリポジトリ上に存在しない
編集	既存のファイルの中身を変化させる	リポジトリ上のファイルパスが講師情報に存在するがキャッシュ値が異なる

2.1.2 ファイル操作の組み合わせ

サーバーリポジトリ上のファイルとクライアントリポジトリ上のファイルそれぞれに対してファイル操作が想定される。
ファイル操作のすべての組み合わせに対して、ユースケースとしてありうるか、コンフリクトが発生するかという観点で以下のようにまとめられる。

ファイルの操作		ユースケース		同期処理		コンフリクト	
サーバー	クライアント	発生有無※１	識別名	発生有無※２	識別名	発生有無※3	識別名
操作なし	操作なし	○	Non	×	-	×	-
追加	操作なし	○	Add	○	SYNC_A	×	-
削除	操作なし	○	Del	○	SYNC_D	×	-
編集	操作なし	○	Edi	○	SYNC_E	×	-
操作なし	追加	○	Add	○	SYNC_A	×	-
追加	追加	○	AddAdd	△	SYNC_AA	△※4	CONF_AA
削除	追加	×	-	-	-	-	-
編集	追加	×	-	-	-	-	-
操作なし	削除	○	Del	○	SYNC_D	×	-
追加	削除	×	-	-	-	-	-
削除	削除	○	DelDel	○	SYNC_DD	×	-
編集	削除	○	DelEdi	×	-	○	CONF_DE
操作なし	編集	○	Edi	○	SYNC_E	×	-
追加	編集	×	-	-	-	-	-
削除	編集	○	DelEdi	×	-	○	CONF_DE
編集	編集	○	EdiEdi	△	SYNC_EE	△※4	CONF_EE

※１ユースケース派生有無の記号の意味
○：発生する
×:発生しない

×：発生しない
※2ユースケース派生有無の記号の意味
○:発生する
△：場合により発生する
×:発生しない
-:ユースケースなし

※3コンフリクト発生有無の記号の意味
○:発生する
△：場合により発生する

×:発生しない
-:ユースケースなし

※4コンフリクト発生の条件
２つのリポジトリのファイルのキャッシュ値が異なればコンフリクト発生
キャッシュ値が同じであれば同期処理を実施する

2.1.3 ユースケースごとの処理

2.1.3.1 ユースケース

識別名	操作概要
Non	２つのリポジトリ内と構成情報で一貫性が保たれているまま
Add	片方のリポジトリに新たにファイルが追加された
Del	片方のリポジトリからファイルが削除された
Edi	片方のリポジトリのファイルが編集された(内容が変更された)
AddAdd	2つのリポジトリに同じパス名のファイルが追加された
DelDel	２つのリポジトリから同じファイルが削除された
DelEdi	既存のファイルを片方のリポジトリから削除され、片方のリポジトリは編集された
EdiEdi	２つのリポジトリのファイルが編集された(内容が変更された)

2.1.3.2 同期処理

識別名	処理内容(構成情報)	処理内容(ファイル)
SYNC_A	構成情報を追加する	ファイルを追加(複製)する
SYNC_D	構成情報を削除する	ファイルをトラッシュフォルダに移動する
SYNC_E	構成情報を更新する	編集されていないリポジトリのファイルをトラッシュフォルダに移動し、 対リポジトリのファイルを複製する。
SYNC_AA	構成情報を追加する	-
SYNC_DD	構成情報を削除する	-
SYNC_EE	構成情報を更新する	-

2.1.3.3 コンフリクト処理

コンフリクト発生検出時はファイルはそのままにしコンフリクト情報を更新する。（参照：4.2.2 コンフリクト情報）

コンフリクトの解消処理は以下の表から選択する。

識別名	処理内容
CONF_AA	サーバー側のファイルを残す
	クライアント側のファイルを残す
CONF_DE	ファイルを削除する
	編集されたファイルを残す
CONF_EE	サーバー側のファイルを残す
	クライアント側のファイルを残す

3.システム機能

3.1 機能一覧

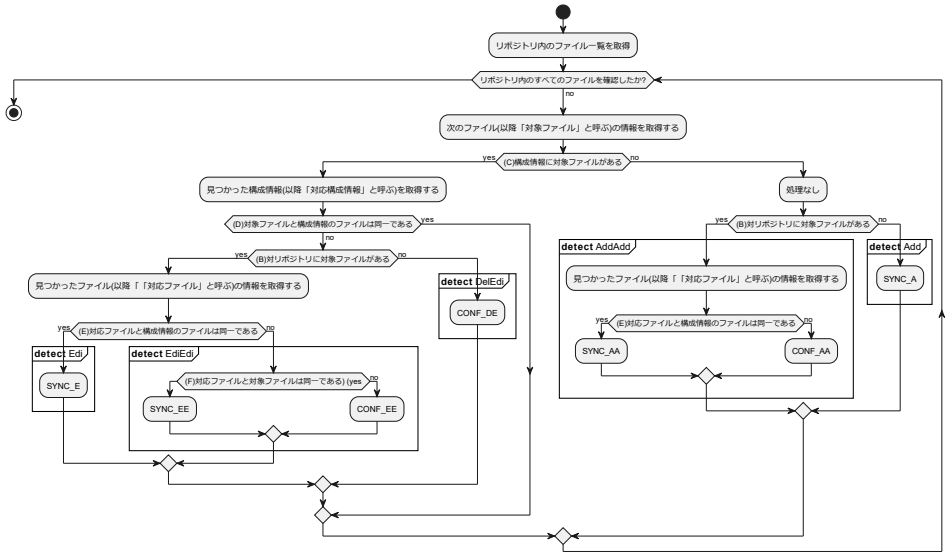
機能名	略称	説明
管理機能	MNG	各機能の呼び出し、結合を管理する機能
差分検出機能	DETECT	リポジトリの変化を検出する機能
同期処理機能	SYNC	リポジトリの同期処理を実行する機能
通知機能	NOTICE	コンフリクトの発生、リポジトリ容量に関する通知を実施し、レスポンスを受け取る機能
データ管理機能	DATA	リポジトリ構成やコンフリクトに関する情報を外部ファイルで管理する機能
シェル操作機能	CMD	OSのシェルを操作しリポジトリ内のファイル操作を実行する機能

3.2 機能設計

3.2.1 差分検出機能

１回の差分検出は以下の順に実行する。
・サーバーリポジトリ差分検出→クライアントリポジトリ差分検出→構成情報差分検出

サーバーリポジトリ差分検出及びクライアントリポジトリ差分検出は以下の条件分岐に従って差分を検出する。



構成情報差分検出は以下の条件分岐に従って差分を検出する。

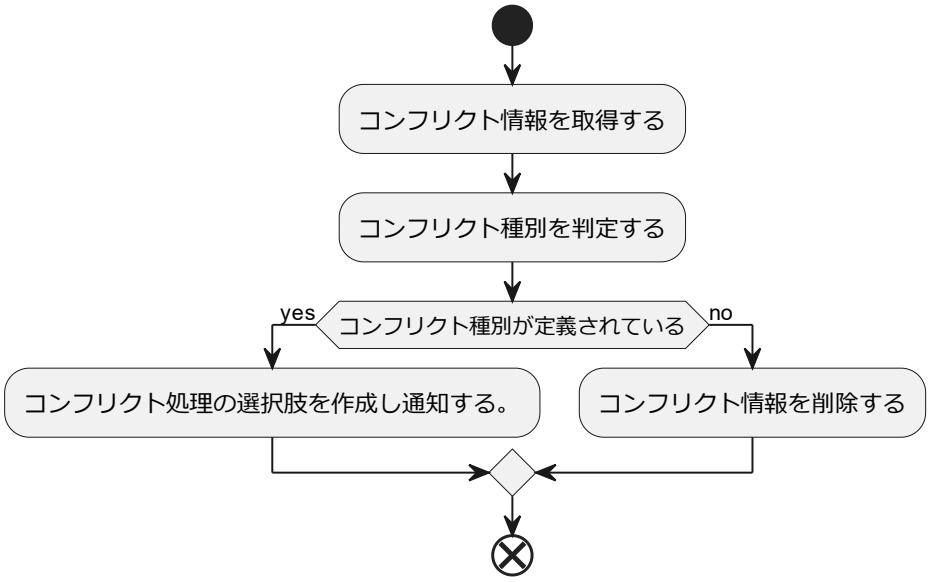
```
EntryNotFound (FileSystemError): Error: ENOENT: no such file or directory, open 'c:\Users\makiy\programing\python\PhotoA
```

3.2.2 同期処理機能

同期処理はファイルパスと処理の種類をインプットにして同期処理を実行する。
処理の概要は 2.1.3.2 同期処理を参照する。

3.2.3 コンフリクト処理機能

コンフリクト情報を取得し、各コンフリクト情報に対して以下の処理を実行する。



4.データ設計

4.1 外部ファイルサイズ制約

各ファイル、フォルダのサイズ上限を以下のように定める

名前	種類	上限サイズ
サーバーリポジトリ	フォルダー	1TB
クライアントリポジトリ	フォルダ	1TB
トラッシュフォルダ	フォルダ	200GB
設定ファイル	ファイル	1GB
構成情報ファイル	ファイル	32GB
コンフリクト情報ファイル	ファイル	32GB
ログファイル	ファイル	1GB

4.2 データ定義

外部ファイルで扱うデータについて定める。

4.2.1 構成情報

構成情報では以下の情報を扱う。

情報	形式	サイズ(上限)	説明
パス	文字列	256バイト	リポジトリの親フォルダからファイル名までのパス
ハッシュ値	文字列	16バイト	SHA256方式で取得したファイルのハッシュ値
サイズ	整数	2バイト	ファイルサイズ(MB単位)
確認日時	文字列	20バイト	構成情報を最後に確認した日時

※パスの表し方　スラッシュ'/'で階層を表す。
※日時の表し方　'YYYY-MM-DD-hh-mm-ss'の形式

4.2.2 コンフリクト情報

コンフリクト情報で扱うデータを設計する。
コンフリクト情報では以下の情報を扱う。

情報	形式	サイズ(上限)	説明
構成情報上のパス	文字列	256バイト	構成情報に保持されているパス
構成情報上のハッシュ値	文字列	16バイト	構成情報に保持されているハッシュ値
サーバーリポジトリ上のパス※	文字列	256バイト	サーバー上のファイルのパス
サーバーリポジトリ上のハッシュ値※	文字列	16バイト	サーバー上のファイルのハッシュ値
クライアントリポジトリ上パス※	文字列	256バイト	クライアント上のファイルのパス
クライアントリポジトリ上のハッシュ値※	文字列	16バイト	クライアント上のファイルのハッシュ値

※サーバー、クライアント上のファイルの情報について削除されている場合はNULL(空欄)とする。

4.2.2.1 コンフリクト発生ケースとコンフリクト情報の対応

	ファイルに対する操作		コンフリクト情報の記載の有無		
識別名	サーバーの操作	クライアントの操作	構成情報上のパス	サーバーリポジトリ上のパス	クライアントリポジトリ上のパス
CONF_AA	追加	追加	×	○	○
CONF_DE	削除	編集	○	×	○
	編集	削除	○	○	×
CONF_EE	編集	編集	○	○	○

○:情報あり
×：情報なし（空欄）

4.2.3 設定情報

設定情報で扱うデータを設計する。
設定情報では以下の情報を扱う。

クラス名	情報	形式	説明
ServerPath	サーバーリポジトリパス	文字列	サーバーリポジトリのOS上での絶対パス
ClientPath	クライアントリポジトリパス	文字列	クライアントリポジトリのOS上での絶対パス
CompositionFilePath	構成情報のファイルパス	文字列	構成情報ファイルのOS上での絶対パス
ConflictFilePath	コンフリクト情報ファイルパス	文字列	コンフリクト情報ファイルのOS上での絶対パス

4.3 帳票設計

外部ファイル(構成情報、コンフリクト情報)の形式、レイアウトを設計する。

4.3.1 構成情報ファイルレイアウト設計

構成情報を読み書きする外部ファイルのレイアウトを設計する。

4.3.1.1 シリアライズフォーマット

csv形式を用いる。
ファイル名は設定ファイルで規定する

4.3.1.2 情報の記載法

ファイルの記載方法を規定する。

以下のように1つの構成情報を１行に記載する。
構成情報の要素をコンマ","で区切り、以下の順番で記載する。

> (パス),(ハッシュ値),(サイズ),(確認日時)

4.3.1.3 制約

・同じパスの情報を複数記載しない
・1行の最大容量は300バイトとする(300バイト*500,000ファイル=32GB)

4.3.2 コンフリクト情報ファイルレイアウト設計

コンフリクト情報を読み書きする外部ファイルのレイアウトを設計する。

4.3.2.1 シリアライズフォーマット

csv形式を用いる
ファイル名は設定ファイルで規定する

4.3.2.2 情報の記載法

ファイルの記載方法を規定する。

以下のように1つの構成情報を１行に記載する。
構成情報の要素をコンマ","で区切り、以下の順番で記載する。

> (構成情報上のパス),(構成情報上のハッシュ値),(サーバーリポジトリ上のパス),(サーバーリポジトリ上のハッシュ値),(クライアントリポジトリ上のパス),(クライアントリポジトリ上のハッシュ値),

4.3.3 設定情報ファイル

設定情報を読み書きする外部ファイルのレイアウトを設計する。

4.3.3.1 シリアライズフォーマット

JSON形式を用いる
ファイルはソースファイルと同じフォルダ下に配置する。
ファイル名は config.json とする

4.3.3.2 情報の記載法

"4.2.3 設定情報"を参照に各設定値を記載する。

5.インターフェイス設計

それぞれの機能が提供するAPIを設計する。

5.1.管理機能API

5.1.1.同期開始

要素	内容
機能概要	外部から同期処理開始要求を受け付けるAPI。APIを呼び出されたとき順次同期処理、通知処理を実施する。
関数名	mng_begin
引数	none
戻り値	none
備考	-

5.2.データ管理機能API

5.2.1.設定情報読み込み機能

要素	内容
機能概要	設定情報ファイルから設定情報を読み出す
関数名	data_read_config
引数	none
戻り値	設定データ
備考	-

5.2.2.ログ書き出し機能

要素	内容
機能概要	1つのログ情報をログファイルに書き出す。
関数名	data_write_log
引数	ログ情報
戻り値	実行結果
備考	-

5.2.3.構成情報追加機能

要素	内容
機能概要	構成情報を追加する
関数名	data_write_composition
引数	構成情報
戻り値	実行結果
備考	同じファイルパスがある場合は追加処理を実行せず、戻り値で通知する

5.2.4.構成情報削除機能

要素	内容
機能概要	構成情報を削除する
関数名	data_remove_composition
引数	ファイルパス
戻り値	実行結果
備考	対象ファイルパスの構成情報がない場合はnot foundを返す

5.2.5.構成情報検索機能

要素	内容
機能概要	構成情報を取得する
関数名	data_read_composition
引数	ファイルパス
戻り値	構成情報
備考	ファイルパスに一致する構成情報がある場合はその構成情報を返す。ない場合はNULLを返す

5.2.5.構成情報検索機能

要素	内容
機能概要	構成情報を取得する
関数名	data_read_composition
引数	ファイルパス
戻り値	構成情報
備考	ファイルパスに一致する構成情報がある場合はその構成情報を返す。ない場合はNULLを返す

5.2.6.コンフリクト情報書き出し機能

要素	内容
機能概要	コンフリクト情報を書き出す
関数名	data_write_conflict
引数	コンフリクト情報
戻り値	実行結果
備考	コンフリクト情報は1つずつ書き出す

5.2.7.コンフリクト情報読み込み機能

要素	内容
機能概要	コンフリクト情報を読み込む
関数名	data_read_conflict
引数	none

要素	内容
戻り値	コンフリクト情報
備考	-

5.3.差分検出機能API

5.3.1

5.3.2

5.4.同期処理機能API

5.5.通知機能API

6.ER図

7.フローチャート