

参照プロジェクトと構成管理ツール によるチームモデリング

■目的:

表題の方法について説明します

■対象:

- ・はじめてチームモデリングの運用を考えられている方
- ・JUDE/Serverからの移行を検討される方



チェンジビジョン
JUDE開発部
2007.8.9

Seeing is understanding.

チームモデリングの課題

- 課題
 - ひとつのファイルだけでモデリングする場合の課題
 - 同時に一人しか編集できない
 - 常に全体のモデルを開く必要がある(性能、焦点)
 - 複数のファイルに分けてモデリングする場合の課題
 - 整合性維持のために、こまめに手でマージする必要がある
 - 別ファイルで管理されているモデルを参照側で編集してしまい、整合性が崩れる場合がある
 - いつ誰がどんな変更をしたのか把握したい
- これらの課題に対するご提案
 - Pro5.0で追加された**参照プロジェクト機能**を活用
 - **構成管理ツール**の活用

「参照プロジェクト＋構成管理ツール」 の運用イメージ

1. 全体イメージ
2. 参照プロジェクトの設定
3. 構成管理の開始
4. 修正とコミット
5. 修正の取り込みと参照プロジェクトの更新
6. コミット履歴とバージョン管理

1. 全体イメージ

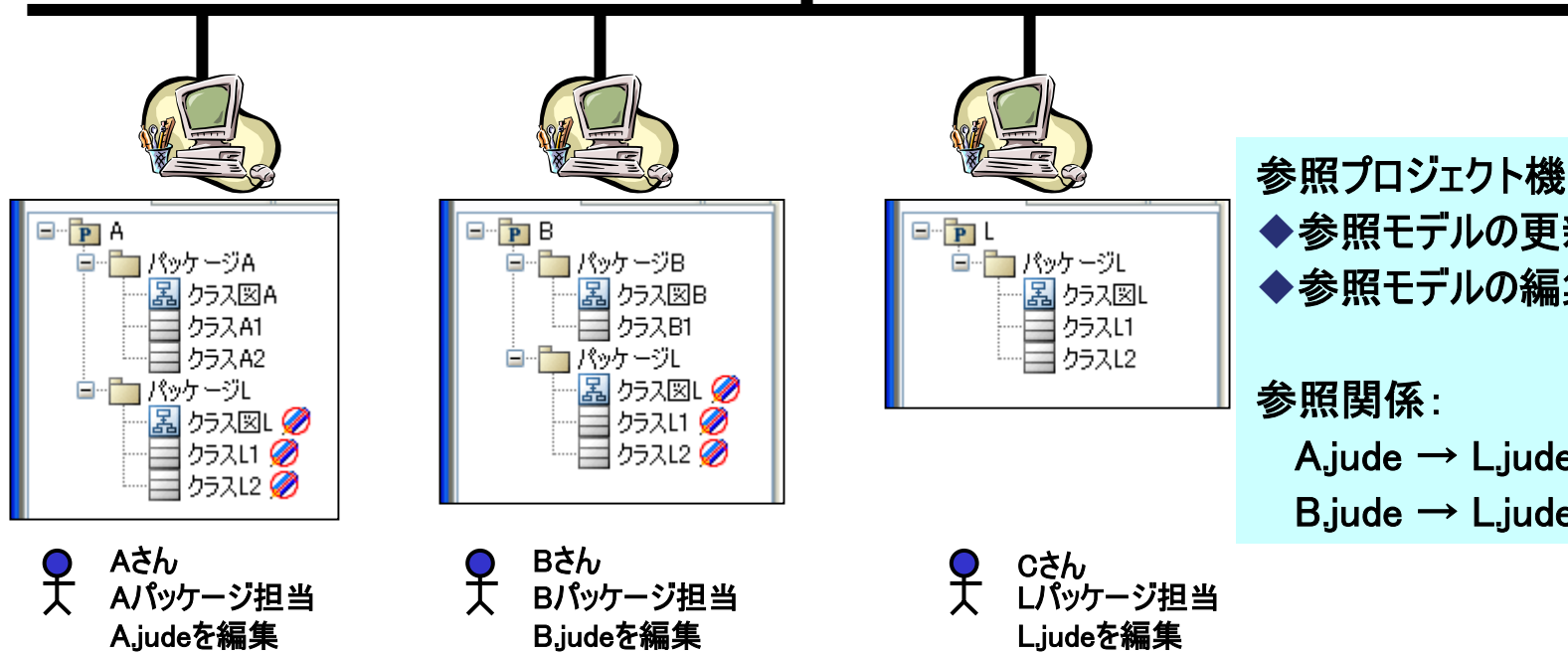
構成管理サーバ



構成管理ツール:

CVSやSVN、VSSなど

- ◆各バージョンのデータ
- ◆修正履歴
- ◆ロック情報 など



参照プロジェクト機能:

- ◆参照モデルの更新補助
- ◆参照モデルの編集禁止

参照関係:

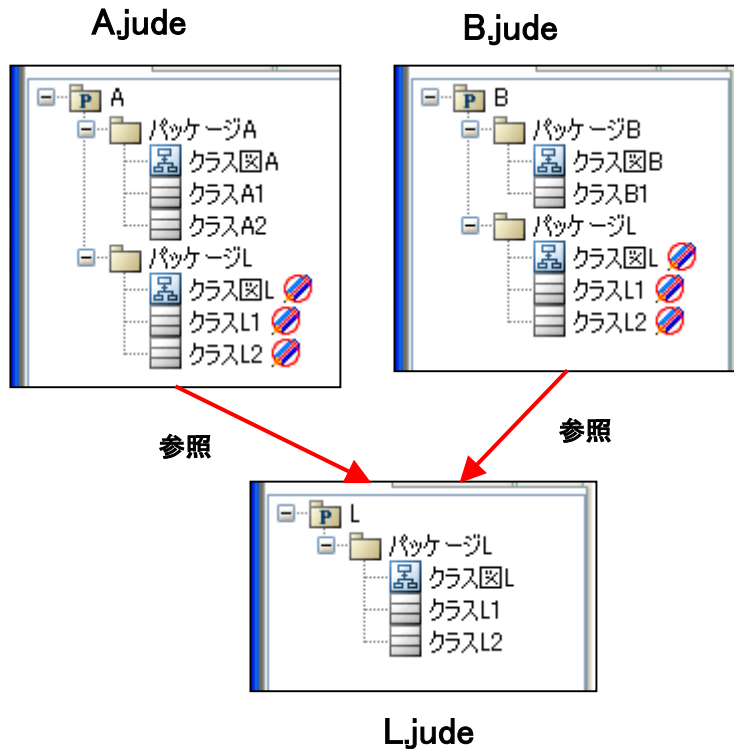
- Ajude → Ljude
- Bjude → Ljude

● Aさん
Aパッケージ担当
Ajudeを編集

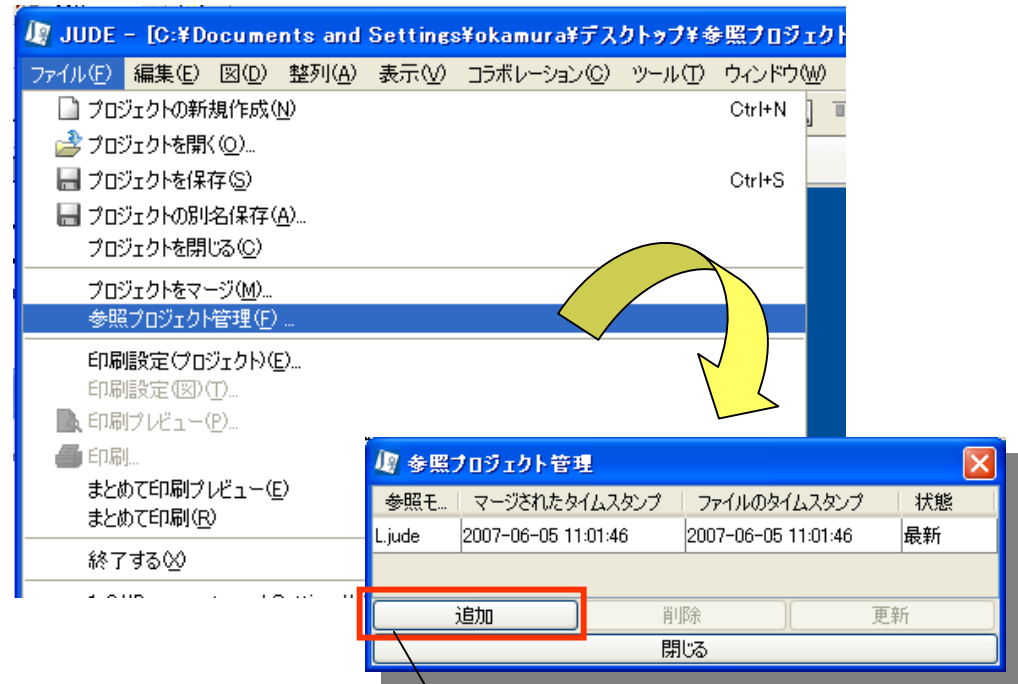
● Bさん
Bパッケージ担当
Bjudeを編集

● Cさん
Lパッケージ担当
Ljudeを編集

2. 参照プロジェクトの設定



Ajudeを開いて「ファイル」メニューから参照プロジェクトとして、Ljudeを登録します



Ljudeを選択して追加

Bjudeについても同様に、Ljudeを参照プロジェクトとして登録します。

Seeing is understanding.

3. 構成管理の開始

1. 構成管理ツールが使える環境を準備
 - CVS、SVN、VSSなど
 - サーバとクライアント
2. JUDEのファイル群をプロジェクトとして登録
3. そのプロジェクトに対してロックなどを使うように設定
 - バイナリファイルのため、同一ファイルの同時編集を避けます
 - 注) 方法は構成管理ツールにより設定が異なります
4. 各クライアントでそのプロジェクトをチェックアウト

※各構成管理ツールの使い方などについては、各構成管理ツールをご確認ください。

4. 修正と修正の反映

Aさんが、Ajudeを開いて、編集します

1. Ajudeのロックを構成管理ツールで取得
2. JUDEで開いて、クラスA3を追加し、保存

- ✧ 参照プロジェクト中のモデル
 - ✧ 編集不可なのでミス抑制
 - ✧ 属性の型などとして参照は可能

3. 構成管理ツールで、Ajudeをコミット(修正の反映)

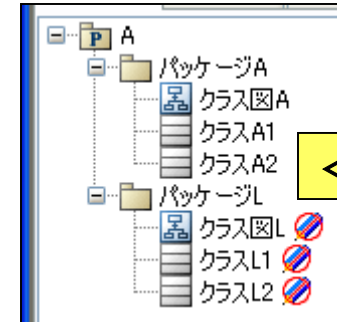
- ✧ コミットのコメントとして編集内容を記入
- ✧ ロックは状況により解放

この間、並行してCさんが、Ljudeを編集します

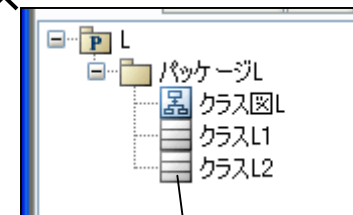
上と同様に

- ✧ LjudeのクラスL2をクラスNEWにリネーム
- ✧ コミット

人 Aさん Ajude



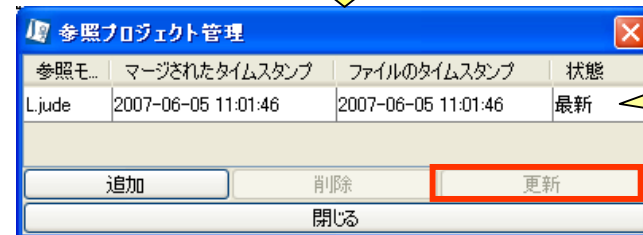
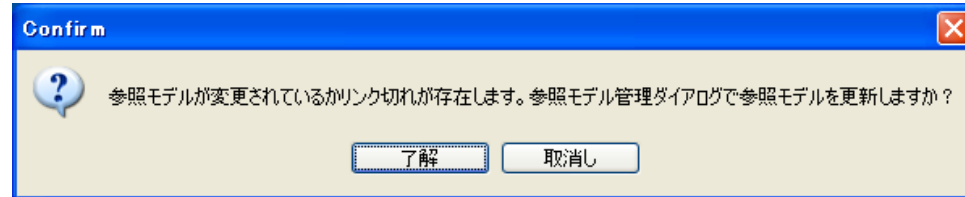
人 Cさん Ljude



5. 修正の取り込みと参照プロジェクトの更新

Bさんが変更を取り込みます

1. 構成管理ツールで、変更を確認
2. 変更されたAjudeとLjudeをアップデート(修正の取り込み)
3. BjudeをJUDEで開く
 - ◇ 参照プロジェクトが更新されている旨の通知表示
4. 参照プロジェクトの更新
 - ◇ 「参照プロジェクト管理」ダイアログで“要更新”のLjudeを更新
 - ◇ クラスL2の名前が、クラスNEWに変わります
 - ◇ クラスB1が、属性:クラスL2を持っている場合、属性:クラスNEWに更新されます



- 最新
- 要更新
- リンク切れ

※参照プロジェクトの更新では、すべて参照プロジェクト側の修正を優先します

6. コミット履歴とバージョン管理

構成管理ツールの機能により、

- 編集履歴を参照可能
- 古いバージョンを取得可能
 - 参照プロジェクトの関係があっても、各ファイルが変更された時と同様に更新されます

日時	リビジョン番号	ユーザ	コメント	修正ファイル群
2007.7.25 14:10	5	Bさん	Lの変更の反映	B.jude
2007.7.25 12:05	4	Lさん	クラスL1をクラスLLLにリネーム	L.jude
2007.7.23 9:30	3	Aさん	クラスA3を追加	A.jude
2007.7.9 15:20	2	Xさん	。。。。。。	。。。。
2007.7.7 10:00	1	Zさん	プロジェクトの登録	。。。。



まとめ：参照プロジェクト機能＋構成管理ツール

機能	普通のマージのみ	参照プロジェクト＋構成管理	JUDE/Server
ファイルの分割管理	複数ファイル/1プロジェクト	複数ファイル/1プロジェクト	1ファイル/1プロジェクト
ユーザ・グループ管理	なし	構成管理のユーザ	○
参照関係管理	×	○	×
参照モデルの編集禁止	×	○	同時編集をマージ
参照プロジェクトの更新通知	×	○	?
部分チェックイン・チェックアウト	×	○ ファイル単位	○ パッケージ単位
権限管理	ファイル単位？	△ プロジェクト・ファイル単位	○ プロジェクト・パッケージ単位
ロック管理	ファイル単位 (JUDE)	ファイル単位 (構成管理)	パッケージ単位
変更履歴参照	×	○	○
古いバージョンの取得	×	○	×
バックアップ (スナップショット)	-	-	○
その他	サポート対象 不整合がおきやすい	サポート対象 汎用構成管理ツール	サポート対象外 独自サーバ 規模と性能注意

Seeing is understanding.

参考情報

- 階層をもつ参照プロジェクトの一括更新サポート
 - トータルマージユーティリティ
 - BAT又はSHファイルで、対象となるJUDEファイルを一括更新してくれるコマンドラインから実行するもの。自動的にプロジェクト間の参照関係を把握して更新します。
- .judeのモデルの比較
 - XML出力して比較
 - マージの詳細ダイアログで差異の確認
- 参照プロジェクト＋構成管理なしの運用
 - 構成管理ツールを使わずに、共有ファイルを利用
 - 共有ファイルの権限で権限管理