ASTERIA WARP LoopEndコンポーネントの不具合について

The Information Cafeteria

infoteria
The Information Cafeteria

asteria warp
Business Automation Platform

この資料の趣旨

2007年11月 ASTERIA WARPの全バージョンで

- 呼び出し元フローがトランザクション化されておらず、
- サブフローのみがトランザクション化されており、
- サブフロー内でLoopEndコンポーネントが使用されている場合

にLoopEndコンポーネントの実行時に、トランザクションがコミットされるという不具合が発見されました。

この不具合はバージョン4.1.1(ビルド番号4.1.1.1475)にて修正されましたが、この修正により既存のフローの動作が変わる可能性があります。
この資料では不具合の影響を正確に説明するために、まずASTERIAのトランザクションの概要について説明し、その後この不具合が再現するパターンについて説明いたします。

©2007 インフォテリア株式会社

ASTERIA WARP
トランザクションの概要

The Information Cafeteria

infoteria
The Information Cafeteria

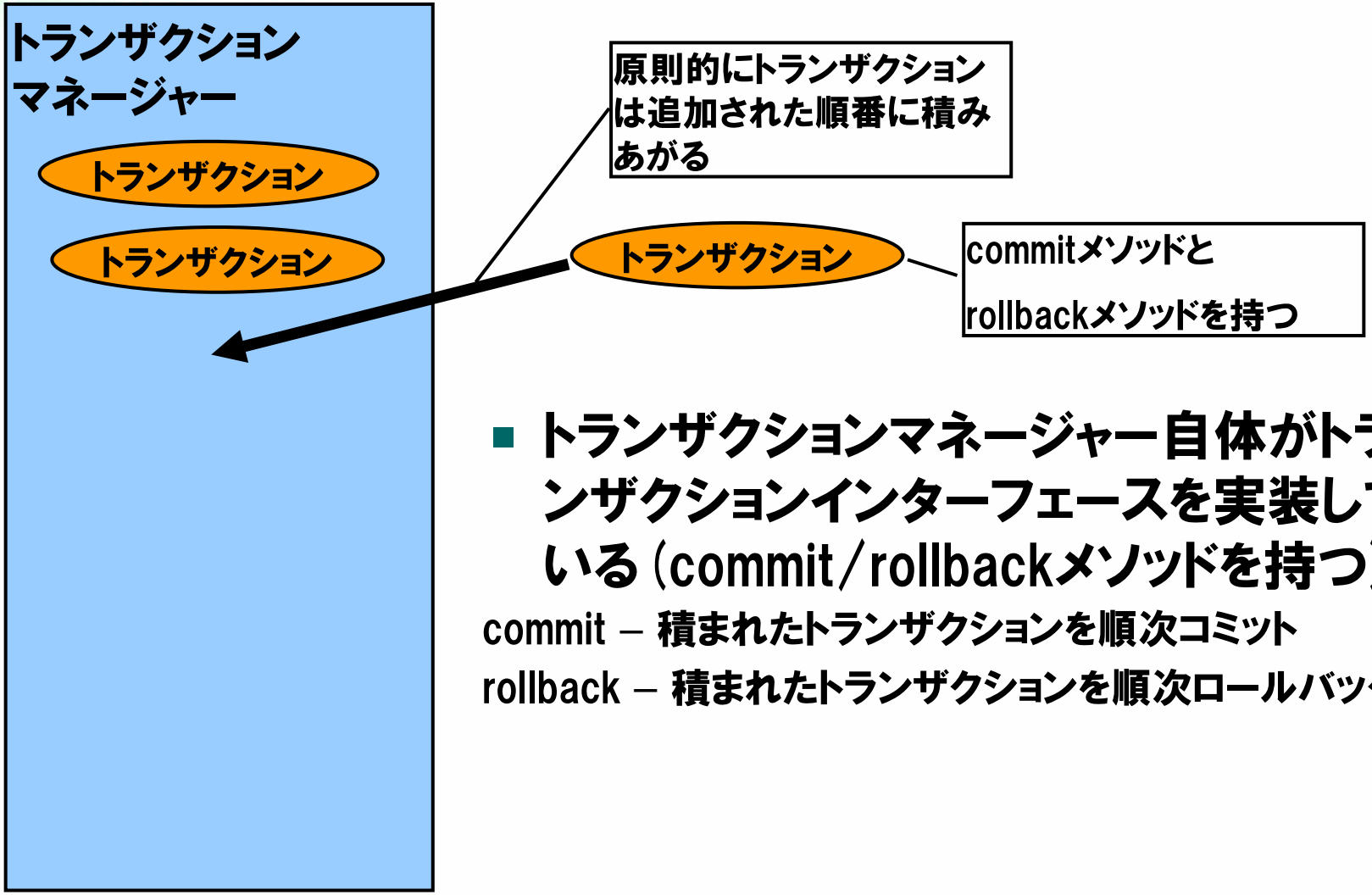
asteria warp
Business Automation Platform

トランザクションについて

- フローのトランザクションはトランザクションマネージャーによって管理される
- リクエストの実行はひとつのトランザクションマネージャーによってハンドルされる（サブフロー、Exceptionフロー、Nextフローすべて）
- トランザクションをサポートするコンポーネントは実行時にトランザクションをトランザクションマネージャーに積む
- トランザクションマネージャーはそのコミットまたはロールバックのタイミングで積上げられたトランザクションをすべてコミットまたはロールバックする
- トランザクションマネージャーがコミットまたはロールバックされるタイミングはフローのトランザクション状態によって決まる
- トランザクションマネージャーに同じトランザクションインスタンスが複数回追加された場合は1度しか積みあがらない。（RDBに対するトランザクションなど）
- コネクションに対するトランザクションは他のトランザクションに比べて優先順位が高い

©2007 インフォテリア株式会社

トランザクションマネージャーの概念図



- トランザクションマネージャー自体がトランザクションインターフェースを実装している（commit/rollbackメソッドを持つ）

commit – 積まれたトランザクションを順次コミット

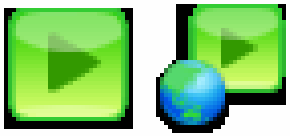
rollback – 積まれたトランザクションを順次ロールバック

©2007 インフォテリア株式会社

フローのトランザクション状態

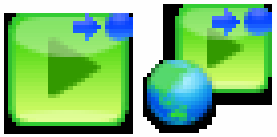
- トランザクション状態はStartコンポーネントの「トランザクション化」プロパティによって決まる

- トランザクション化＝いいえの場合



コンポーネントをひとつ実行するごとにトランザクションマネージャーをコミットする。

- トランザクション化＝はいの場合



アイコンにマークがつく

フローの実行が終了した時にトランザクションマネージャーをコミット（またはロールバック）する。
（コミットするかロールバックするかは終了コンポーネントの「トランザクション」プロパティによって決まる。）

トランザクション状態とサブフロー

- 呼び出し元フローがトランザクション化されている場合は、そこから呼び出されるサブフロー、Exceptionフローのトランザクション化状態はそのStartコンポーネント#トランザクション化の値に関わらずトランザクション化される
- 呼び出し元フローがトランザクション化されていない場合は、そこから呼び出されるサブフロー、Exceptionフローだけをトランザクション化することができる

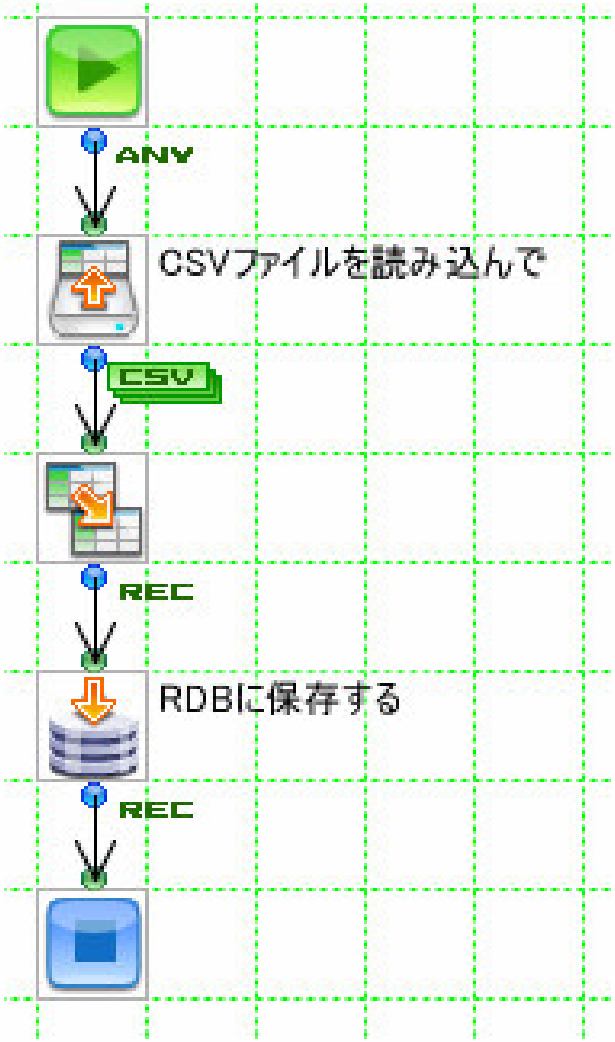
©2007 インフォテリア株式会社

トランザクションをサポートするコンポーネント

- 現在はトランザクションを使用する処理はほとんどRDB系のコンポーネントのみ

©2007 インフォテリア株式会社

トランザクション＝いいえの場合

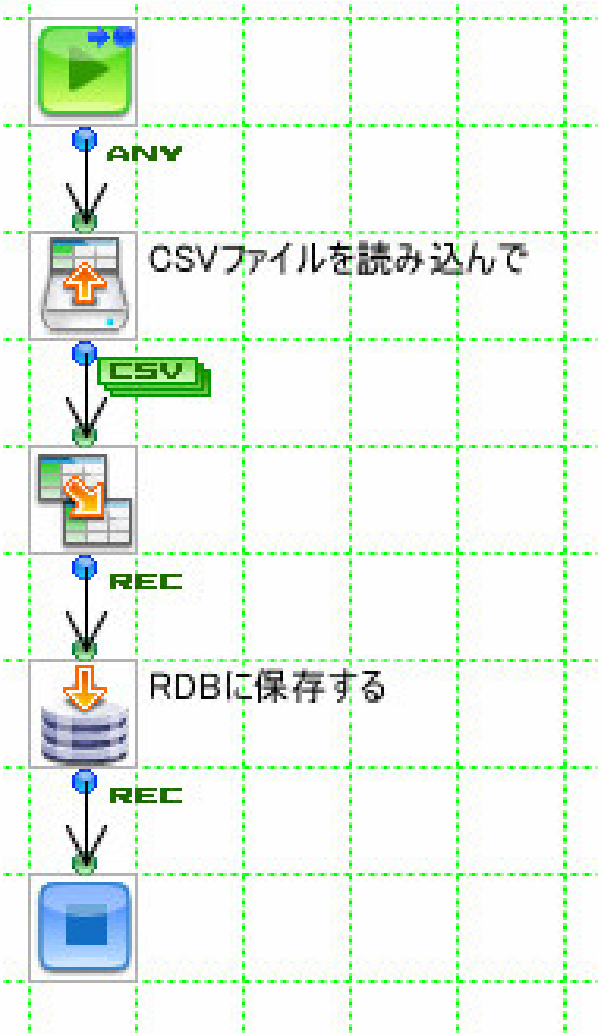


RecordGetでCSVファイルを1行ずつ読み込んでRDBに保存する。
このフローではRDBPutのみがトランザクションをサポートする。(RDBに対するcommitまたはrollback)

1, Start
2, RecordGet
3, Mapper
4, RDBPut (トランザクションを積む)
5, トランザクションコミット
6, 2～5の繰り返し
7, End

©2007 インフォテリア株式会社

トランザクション＝はいの場合



1, Start

2, RecordGet

3, Mapper

4, RDBPut (トランザクションを積む)

5, 2～5の繰り返し

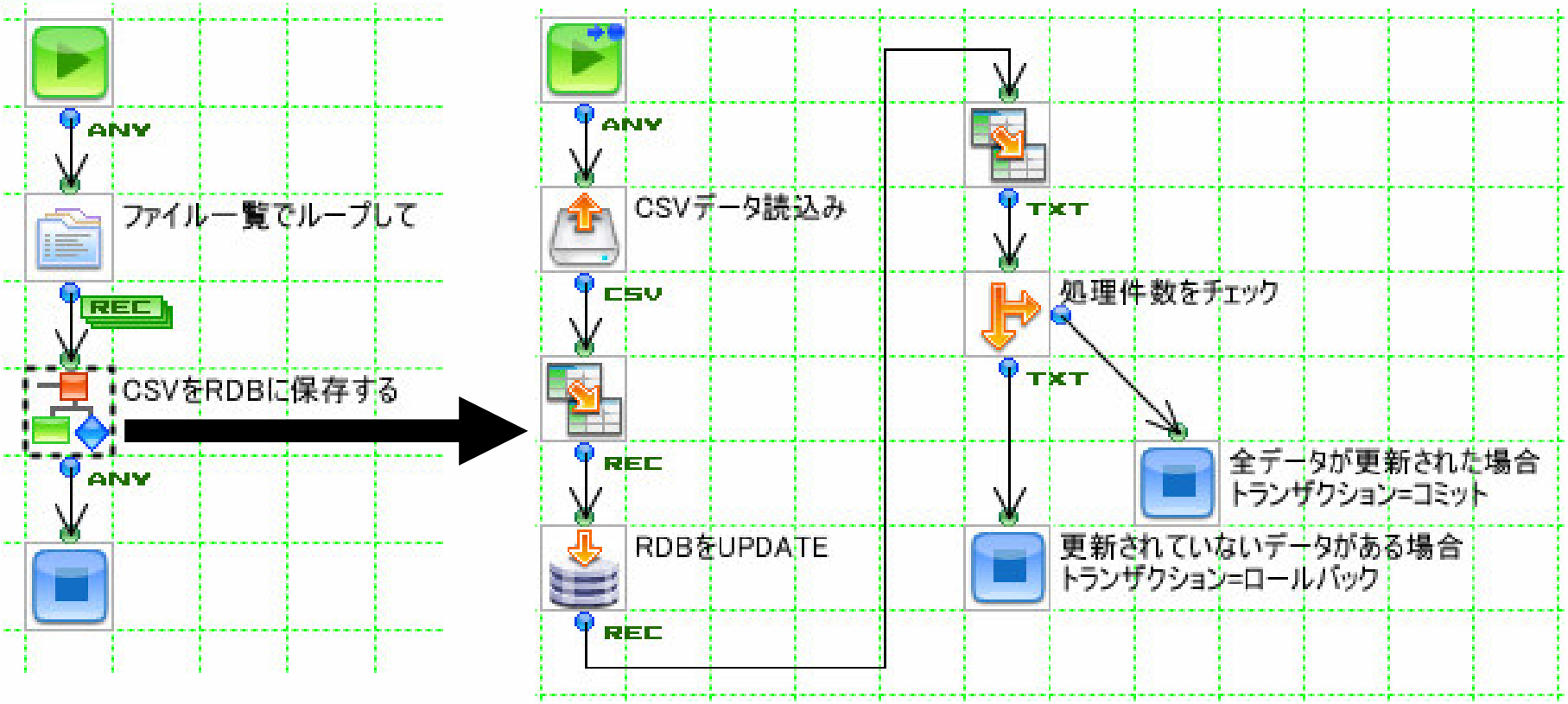
2回目以降のRDBPutで積まれるトランザクションは初回と同じインスタンスなのでトランザクションマネージャーには積みあがらない

6, End

7, トランザクションコミット

©2007 インフォテリア株式会社

サブフローのトランザクション



- サブフローだけがトランザクション化される場合は、条件分岐してコミットするかロールバックするかを選択することができる

今回の不具合について

infoteria
The Information Cafeteria

asteria warp
Business Automation Platform

概要

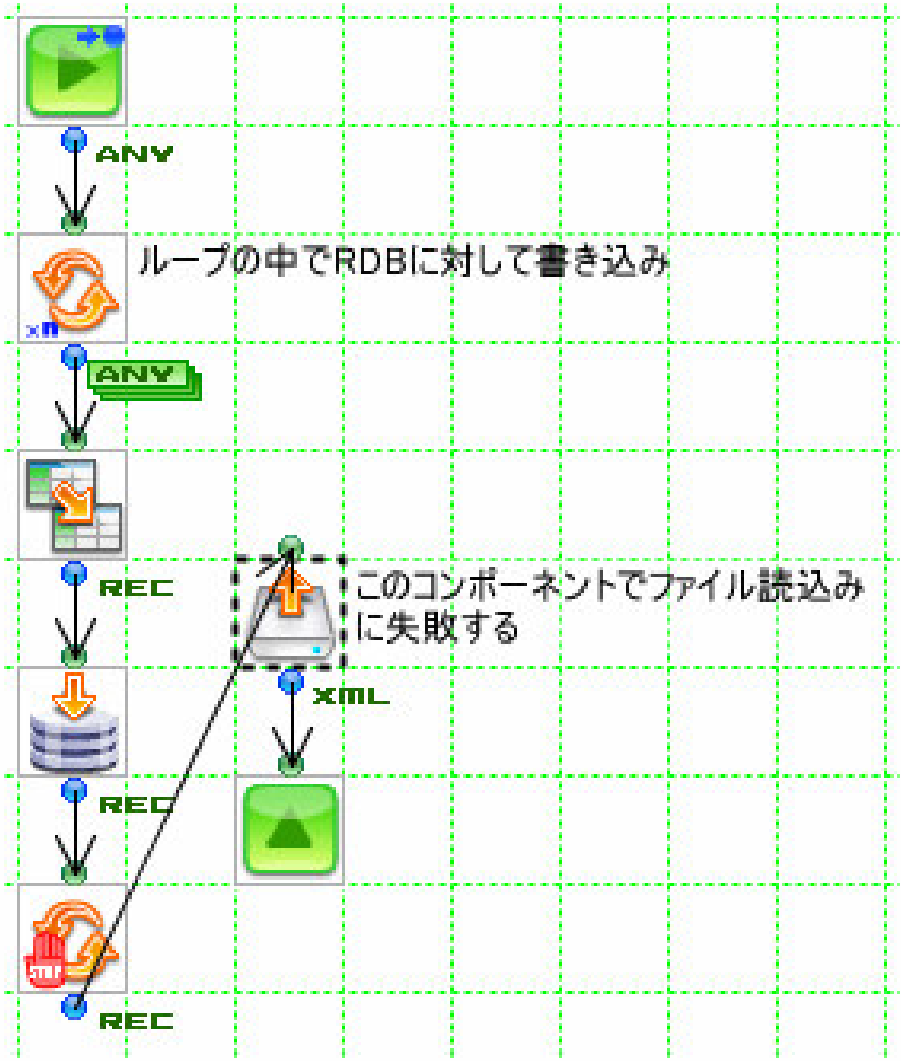
- 呼び出し元フローがトランザクション化されておらず、
- サブフローのみがトランザクション化されており、
- そのサブフロー内でLoopEndコンポーネントが使用されている場合に、
- LoopEndコンポーネント実行時にトランザクションマネージャーがコミットされます。

つまりこの不具合が再現するフローではフロー開発者の意図に反してフローのトランザクションが途中でコミットされます。
この不具合が再現するのは上記条件すべてを満たしている場合です。
つまり次のようなケースではこの不具合は発生しません。

- フローがトランザクション化されていない場合
- メインフローがトランザクション化されている場合。
- LoopEndコンポーネントを使用していない場合

©2007 インフォテリア株式会社

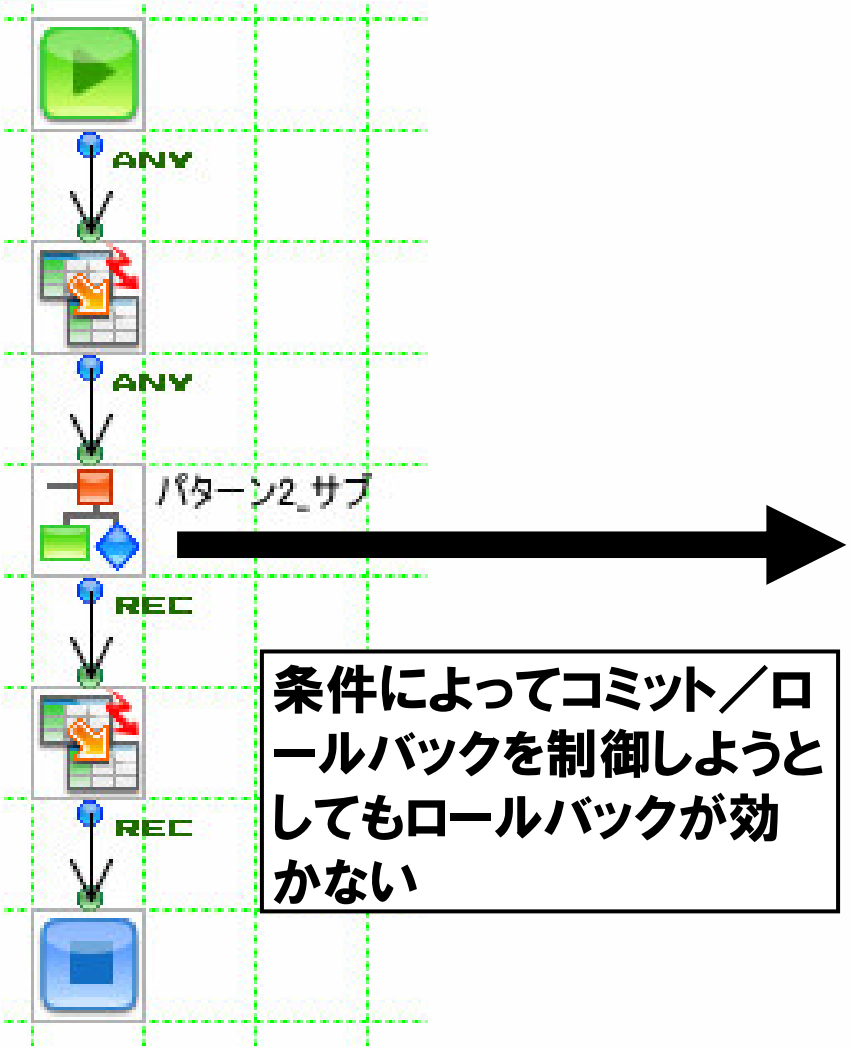
不具合発生の例（1）



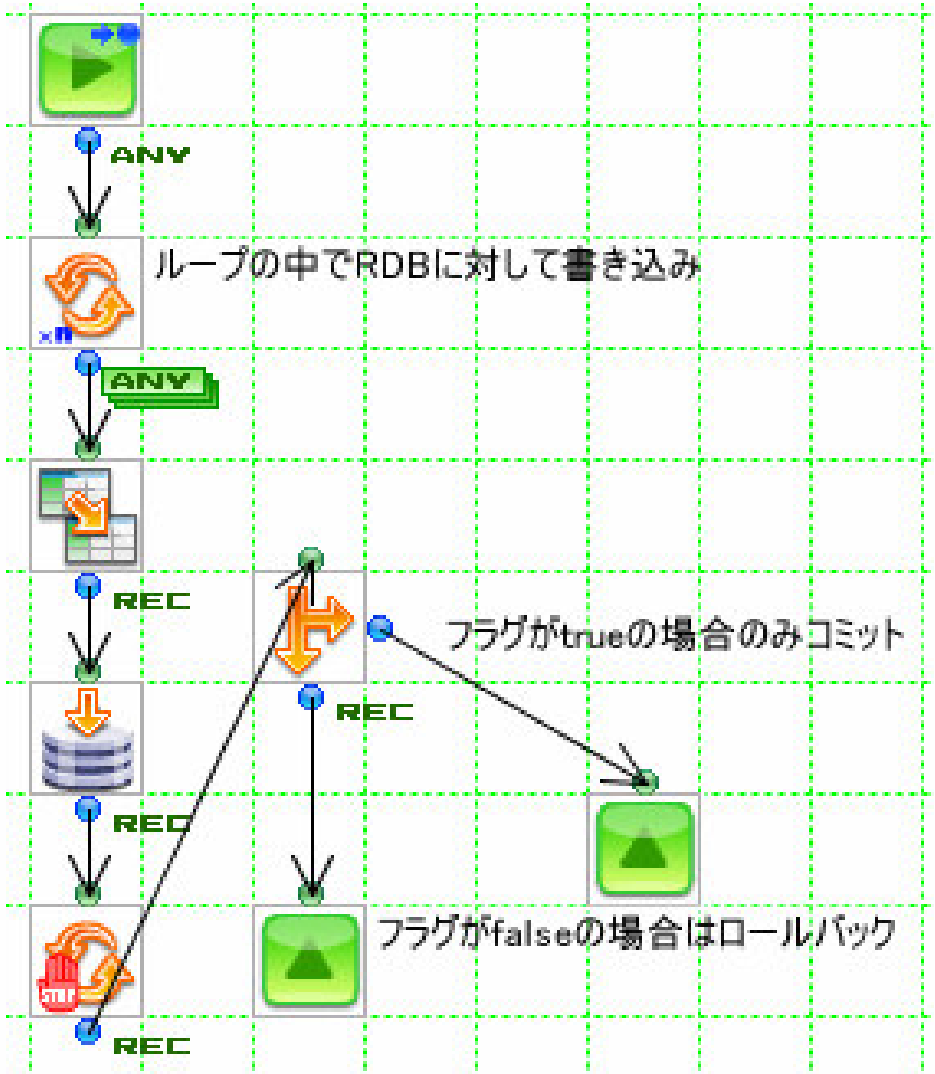
トランザクション化されたサブフロー内でエラーが発生しているので本来ならRDBへの書き込みはロールバックされなければならないがLoopEndコンポーネントでコミットされているのでRDBへの書き込みが行われてしまう。

©2007 インフォテリア株式会社

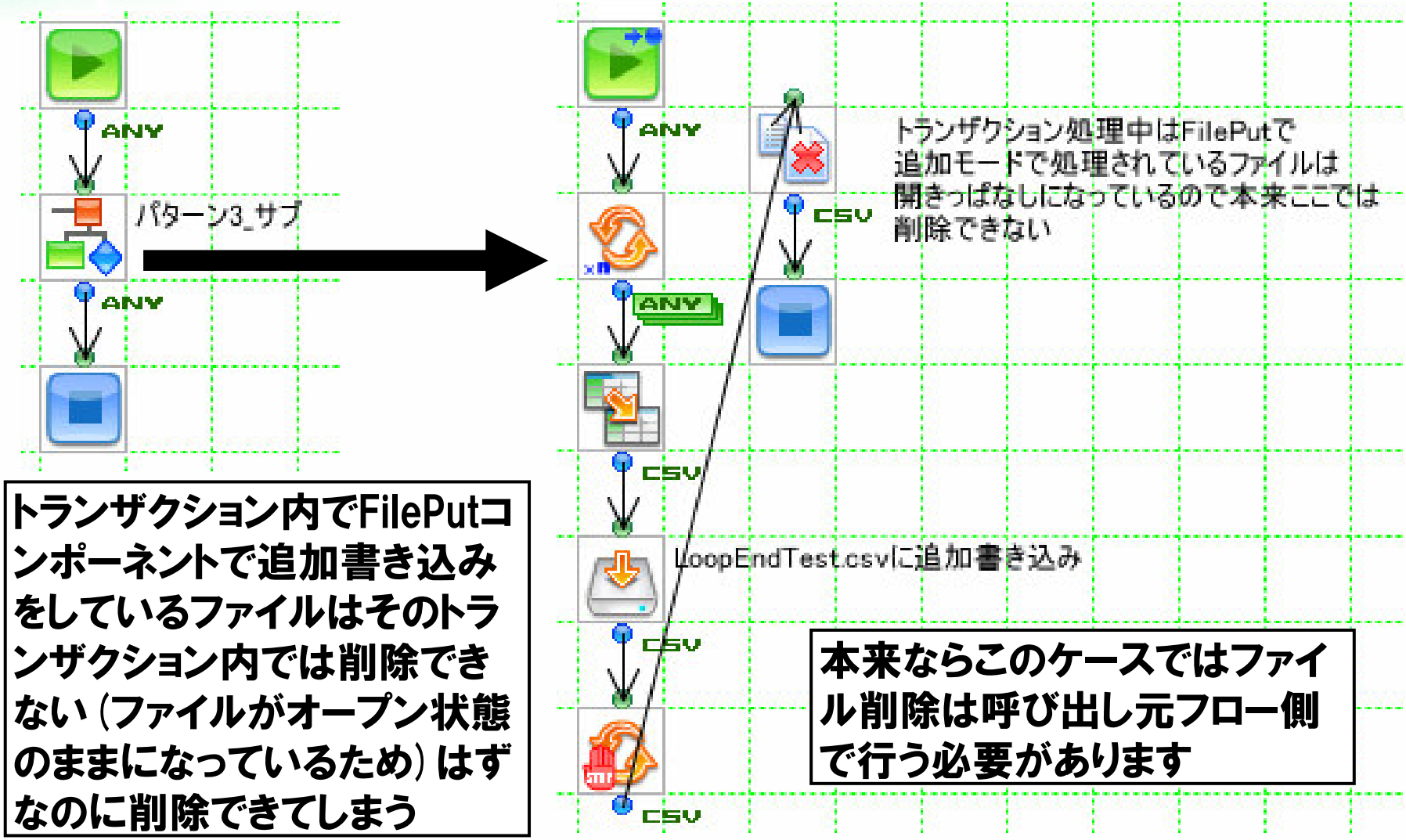
不具合発生の例 (2)



条件によってコミット／ロールバックを制御しようとしてもロールバックが効かない



不具合発生の例（3）



トランザクション内でFilePutコンポーネントで追加書き込みをしているファイルはそのトランザクション内では削除できない（ファイルがオープン状態のままになっているため）はずなのに削除できてしまう

本来ならこのケースではファイル削除は呼び出し元フロー側で行う必要があります

©2007 インフォテリア株式会社

まとめ

- この不具合の修正によって既存のフローのトランザクションの動作が変わってしまいます。
- 例1のケースでは開発の際にトランザクションの動作不正に気づかないまま運用に入ってしまっているケースもあると考えられます。パッチを適用することで正しい動作に修正されますが、コミットのタイミングが変更されるため現行処理と同等の振る舞いが必要である場合はフローを修正する必要があります。
- 例2のケースでは開発の際にトランザクションの動作不正に気づかれるケースであると考えられます。パッチを適用することで正しい動作に修正されます。
- 例3のケースでは本来はエラーとなるはずのフローがそのまま動作してしまいます。本ケースも例1同様動作不正に気づかないまま運用に入ってしまっているケースもあると考えられます。この場合、パッチを適用することでフロー実行時にエラーが発生してしまいますのでフローを修正(ファイル削除のコンポーネントを呼び出し元フロー側に移動)する必要があります。

現状、不具合に無関係であるとしても追加開発でこの不具合に再現してしまう可能性はありえます。
可能な限りパッチを適用していただけるようお願いします。

©2007 インフォテリア株式会社