

## 4 O Modelo de Transações Proposto

Este capítulo tem por objetivo apresentar o *modelo de transações proposto*. Inicialmente, será apresentada uma introdução ao formalismo ACTA usado na formalização do modelo proposto. A proposta do novo modelo de transações será, então descrita, destacando seus axiomas, sua definição conceitual, os estados de suas transações, a formalização do modelo, seu critério de corretude e as suas contribuições.

### 4.1 Framework ACTA

Os modelos de transações convencionais são baseados na noção de *atomicidade*, caracterizando assim as transações atômicas. A atomicidade implica as propriedades de *falha de atomicidade* e *seriação*. A *falha de atomicidade* significa que ou todas as operações da transação são executadas ou nenhuma operação é executada, enquanto a *seriação* significa que transações concorrentes executam sem nenhuma interferência uma nas outras, como se tivessem uma ordem serial para serem executadas. Entretanto, essas propriedades combinam várias noções importantes, tais como [12]:

1. *Visibilidade* - habilidade de uma transação ver o resultado de outra transação enquanto está sendo executada;
2. *Durabilidade* - habilidade da transação de gravar seus resultados no banco de dados;
3. *Recuperação* - habilidade de retornar o banco de dados a algum estado que é considerado correto, em caso de falhas;
4. *Consistência* - exatidão do estado do banco de dados, que é produzida pela confirmação (*commit*) da transação.

A *flexibilidade* que um modelo de transações pode vir a ter depende basicamente da forma como essas quatro características são combinadas.

*Transações complexas* consistem em um conjunto de operações sobre os objetos e/ou em transações complexas. Assim, as *transações complexas* possuem uma rica estrutura interna, diferentemente das transações tradicionais, que possuem apenas um único nível em sua estrutura [12].

A forma como as partes de uma transação são combinadas para formar a *transação complexa* reflete a semântica da aplicação. Tal semântica pode ser explorada em projetos específicos de *controle de concorrência* e *recuperação* dos modelos de transações.

O *controle específico de concorrência* de transações permite a definição de uma nova noção de conflitos entre as operações não-possíveis, com base nas informações disponíveis sobre objetos e seus tipos. Por exemplo, operações invocadas por duas transações podem ser intercaladas como se fossem comutáveis, se a semântica da aplicação permitir que a dependência entre as transações seja ignorada. Essas transações podem não preservar a serialização, mas ainda preservam a consistência, podendo aumentar o desempenho em sistemas de informações mais complexos.

A *recuperação específica* de transações pode ser definida para explorar a semântica da aplicação, no sentido de minimizar os efeitos das falhas das transações. Assim, se reduz o custo da recuperação, pela tolerância parcial de falhas e tanto pela recuperação posterior quanto pelo retorno a uma situação anterior consistente do banco de dados. No caso de a falha acontecer em uma transação componente, a porção na qual ocorreu a falha pode ser isolada, permitindo que o restante da transação seja processado. A porção na qual ocorreu a falha pode ser reexecutada, compensada pela tentativa de execução de uma outra transação ou simplesmente ignorada. Além disso, *transações complexas* suportam o uso de pontos de controle (*checkpoints*) emitidos pelos usuários.

Para melhor identificarmos e captarmos a *semântica das transações complexas* e como *podemos raciocinar sobre as propriedades de concorrência e recuperação das transações complexas*, em [12] e [13] é apresentado o **framework ACTA**<sup>1</sup>, que é um formalismo para capturar propriedades das transações como as relacionadas com *visibilidade*, *falhas de atomicidade (recuperação)*, *durabilidade* e *consistência*.

O framework ACTA permite a formalização de um modelo de transações, captando, principalmente, a *semântica das transações* e especificando *as propriedades de concorrência e recuperação das transações*. Esse framework permite também a *especificação dos efeitos de uma transação*

---

<sup>1</sup>O termo ACTA significa *ações* em latim.

sobre outra transação, bem como os efeitos sobre os objetos do banco de dados [13].

O framework ACTA é utilizado para investigar a especificação, a análise e sínteses *formais* de um modelo de transações. É um formalismo em lógica de primeira ordem, construído com base na *história da transação*, nas dependências entre as transações, na visão da transação, no conjunto de conflitos das transações e nas delegações que devem ser feitas pela transação [14]. O framework ACTA é utilizado na formalização do modelo de transação proposto, e está detalhado no Anexo B.

## 4.2

### O Modelo da Transação Móvel Proposto

Um *banco de dados* é uma entidade que contém todos os objetos compartilhados do sistema. Uma *transação* acessa e manipula os objetos no banco de dados por meio de operações invocadas especificamente para um objeto. O *estado de um objeto* é representado pelo seu conteúdo. Cada objeto tem um tipo, que define o conjunto de operações que provêem um significado único para criar, alterar e examinar o estado de um objeto desse determinado tipo. Presume-se que uma *operação* sempre produz uma saída, retorna um valor, que é um resultado. O resultado de uma operação sobre um objeto depende do estado do objeto [13].

Uma **transação móvel** é uma transação distribuída, na qual alguma parte da computação é executada no equipamento móvel (*host móvel*) e a outra parte em um equipamento fixo (*host fixo*), não-móvel.

O ambiente de banco de dados que será utilizado neste trabalho obedece aos seguintes axiomas:

**Axioma 1:** Um banco de dados localizado no equipamento móvel é um *fragmento* de algum banco de dados localizado na rede fixa, estando seu esquema<sup>2</sup> contido no esquema do banco de dados de onde ele é extraído.

- Seja  $BD_i$  um banco de dados localizado na rede fixa.
- Seja  $BD_i.I-Obj_j(k)$  a  $k$ -ésima instância do objeto  $j$  armazenado no banco de dados  $BD_i$ .
- Sejam  $BD_i.Frag_1, BD_i.Frag_2, BD_i.Frag_3, \dots, BD_i.Frag_m$  fragmentos do banco de dados  $BD_i$ , onde um fragmento  $p$  qualquer do banco de dados  $BD_i$  é definido pela união das instâncias dos vários objetos de  $BD_i$ . Assim, temos que  $BD_i.Frag_p = \bigcup_{j=1}^w (\bigcup_{x=1}^n BD_i.I-Obj_j(x))$

<sup>2</sup>Catálogo que armazena a descrição do banco de dados [21].

- $BD\text{-Móvel}_r = BD_i.Frag_p$  e  $BD\text{-Móvel}_s = BD_i.Frag_q$ , onde  $p \neq q$  e  $r \neq s$ .

**Axioma 2:** Não existe interseção entre as instâncias dos objetos dos bancos de dados móveis.

- $\forall i, j$  onde  $0 < i, j \leq n$ ,  $i \neq j$ , temos que  $BD\text{-Móvel}_i \cap BD\text{-Móvel}_j = \emptyset$

**Axioma 3:** A inicialização dos dados (carga inicial) de um banco de dados móvel  $BD\text{-Móvel}_i$  é sempre feita por um procedimento executado no banco de dados localizado na rede fixa  $BD_i$ .

**Axioma 4:** A sincronização dos dados no sentido *banco de dados fixo-banco de dados móvel* só pode acontecer caso não haja nenhuma transação pendente ou por executar no banco de dados móvel, e será executada por um processo semelhante ao de inicialização dos dados citado no Axioma 3. Além dessa restrição, o banco de dados móvel deve estar bloqueado para o processamento de qualquer tipo de transação, seja ela de leitura ou atualização.

**Axioma 5:** Não pode haver atualizações nos fragmentos dos bancos de dados localizados na rede fixa por nenhuma aplicação em execução na rede fixa quando esses fragmentos estiverem disponibilizados em algum banco de dados móvel, a menos que essa atualização seja oriunda de transações desse banco de dados móvel. Somente será possível essa atualização por procedimentos do administrador do banco de dados, procedimentos estes que deverão atender ao Axioma 4.

**Axioma 6:** A sincronização dos dados *banco de dados móvel-banco de dados fixo* será efetivada pela ressubmissão das transações móveis, executadas e confirmadas no banco de dados móvel, no banco de dados localizado na rede fixa.

**Axioma 7:** O processamento das transações no equipamento móvel é serial e o processamento dessas transações na rede fixa (sincronização móvel/fixa) também é serial, porém controlado na rede fixa.

**Axioma 8:** Um banco de dados, localizado no equipamento móvel, só pode ter o seu esquema modificado por meio de um procedimento

disparado pelo banco de dados do qual ele é extraído. Nenhuma aplicação móvel ou usuário do equipamento móvel pode alterar o esquema do banco de dados local.

**Axioma 9:** Uma transação que executa no banco de dados localizado no equipamento móvel só pode ser *compensada* se o esquema do banco de dados, localizado no equipamento móvel, apresentar alguma diferença em relação ao banco de dados da rede fixa.

**Axioma 10:** Uma transação móvel confirmada (*commit*) no banco de dados localizado no equipamento móvel e que não apresente divergência de esquema com o banco de dados localizado na rede fixa sempre deverá ser confirmada no ambiente da rede fixa.

Esses axiomas são importantes para garantir que as transações processadas e confirmadas nos equipamentos móveis também sejam processadas nos equipamentos localizados na rede fixa e não sejam canceladas indiscriminadamente, causando problemas na consistência dos dados, bem como para esclarecer a forma como essas transações serão processadas no equipamento móvel e na rede sem fio e a forma como os bancos de dados são sincronizados.

Os Axiomas 1 e 2 definem o conteúdo dos bancos de dados móveis utilizados neste trabalho, definindo-os como fragmentos dos bancos de dados localizados na rede fixa que não apresentam interseção em seus dados.

Os Axiomas 3 e 4 tratam da carga inicial do banco de dados móvel e em que situação se pode atualizar os dados desse bancos através de processos de sincronização com os bancos de dados localizados na rede fixa.

Os Axiomas 5 e 6 especificam como os dados residentes nos bancos de dados localizados nos equipamentos da rede fixa e que estão disponibilizados nos respectivos bancos de dados móveis poderão ser sincronizados. Assim, a consistência dos bancos de dados e o sincronismo das atualizações, nos dois ambientes, são garantidos pela execução da própria transação em ambos os ambientes, garantindo os estados consistentes dos bancos de dados, mesmo que haja alguma defasagem de tempo.

O Axioma 7 especifica a forma de concorrência no processamento de transações de um banco de dados móvel e como essas serão processadas quando enviadas para a rede fixa.

Os Axiomas 8 e 9 buscam garantir a compatibilidade dos esquemas dos bancos de dados, principalmente pela dificuldade de mantê-los atualizados em função da possibilidade da existência de inúmeros usuários móveis,

e dos momentos diferentes de conexão desses usuários, para atualizarem os esquemas de seus bancos de dados locais, no caso de evolução desses esquemas.

O Axioma 10 busca garantir que as transações executadas e confirmadas (*commit*), nos bancos de dados localizados nos equipamentos móveis, também serão confirmadas nos bancos de dados da rede fixa, mesmo em caso de divergências semânticas da aplicação. Caso a transação seja cancelada na rede fixa, ela será reenviada para execução pelo equipamento móvel, até que seja processada e confirmada (*commit*) na rede fixa.

É importante ressaltar que, por se estar trabalhando em ambientes disjuntos, em alguns casos, o conteúdo dos bancos de dados pode apresentar diferenças, porém seus estados são consistentes, individualmente.

Outro fato importante é que, embora o processamento das transações em cada banco de dados móvel seja *serial* e a sincronização via o reprocessamento dessas mesmas transações nos bancos de dados localizados na rede fixa também o seja, isso faz com que possamos *garantir o processamento correto das transações nos dois ambientes*, uma vez que os dados disponíveis em um banco de dados móvel não estão disponíveis, também, em outro (*axioma 2*) banco de dados móvel. Essa serialização das transações nos dois ambiente realmente é um fator restritivo do modelo.

#### 4.2.1

##### **Definição Conceitual do Modelo Proposto**

O modelo de transações proposto considera um sistema de banco de dados que pode estar totalmente distribuído. Os dados estão armazenados nos bancos de dados, localizados na rede fixa, podendo existir subconjuntos desses dados armazenados nos equipamentos móveis, através de replicação ou cópia dos dados. A consistência global será sempre mantida nos bancos de dados localizados na rede fixa, enquanto a consistência local dos dados mantidos nos equipamentos móveis será controlada pelos gerentes de transações locais, identificando os seguintes tipos de consistência nos dados:

1. *Dados consistentes* - quando os dados foram atualizados localmente e globalmente;
2. *Dados consistentes apenas localmente* - quando os dados foram atualizados localmente porém ainda não foram atualizados globalmente;

3. *Dados inconsistentes* - quando os dados apresentaram qualquer tipo de problema, seja na atualização local ou global, e sua utilização local pode comprometer os resultados esperados.

Esses níveis de consistência serão mantidos como um atributo do banco de dados local, na granularidade de tabela, e são atualizados pelas transações.

O modelo de transação móvel proposto nesta tese é uma hierarquia em quatro níveis contendo diversos tipos de transações aninhadas. O primeiro nível, ou nó mais alto, é o nó chamado de transação-raiz, e os demais níveis da hierarquia representam as transações componentes ou níveis mais baixos da hierarquia. A figura 4.1 é uma representação da estrutura do modelo de transação móvel proposto.

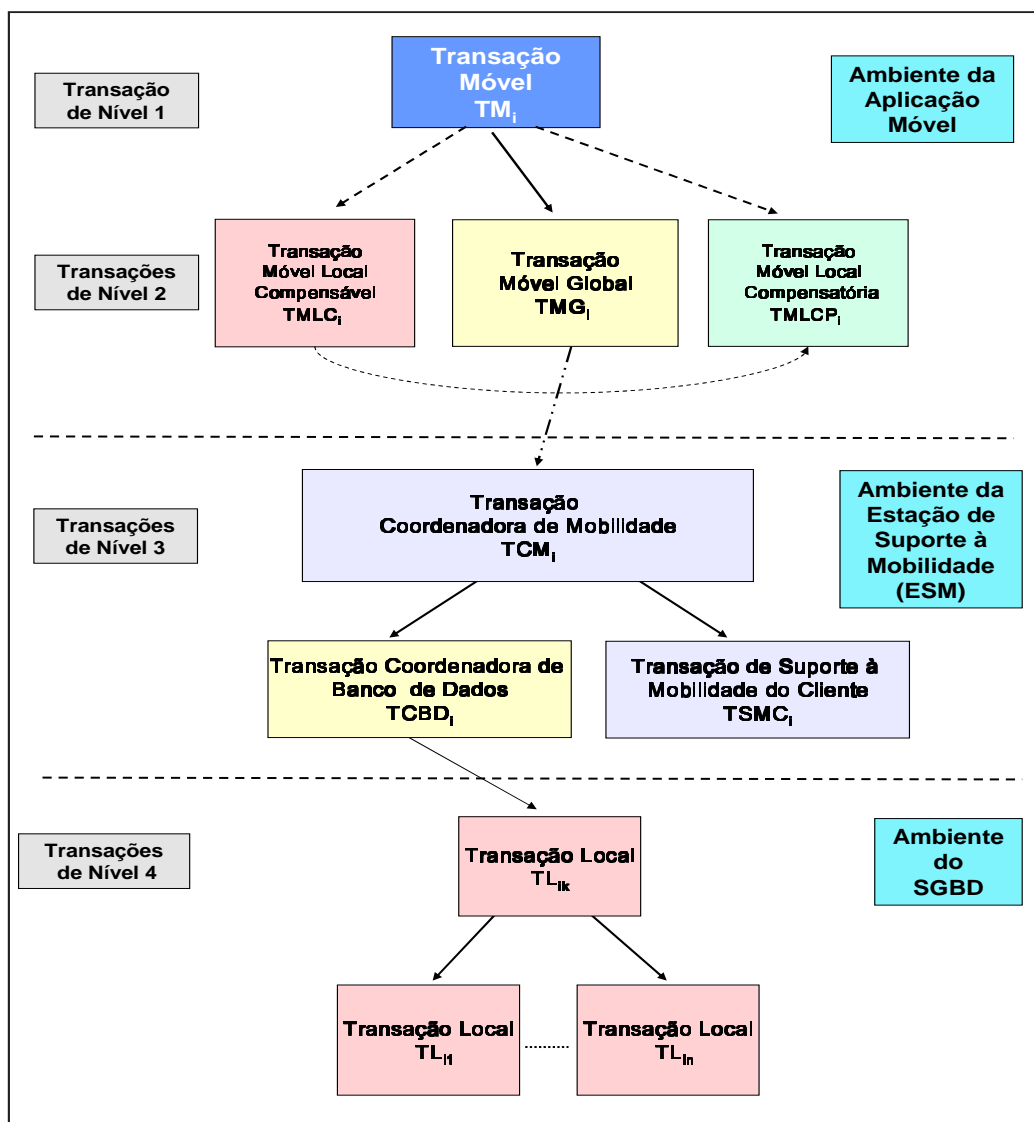


Figura 4.1: Representação do modelo de transações proposto

A concepção dessa hierarquia em quatro níveis deve-se à necessidade de especificar e isolar as funcionalidades do modelo segundo as características e requisitos do processamento das transações, tanto no ambiente sem fio como no ambiente de rede fixa. Além disso, deve assegurar a interatividade entre esses ambientes, visando, principalmente, à garantia da consistência dos bancos de dados. Assim, cada nível possui as seguintes funcionalidades:

- Nível 1 (Aplicação Móvel) - interage com a aplicação residente no equipamento móvel, gera, inicia e termina a Transação Móvel (TM) e controla a execução da TM. Assim, diversos tipos de aplicações podem interagir com o modelo proposto;
- Nível 2 (Banco de Dados Local do Equipamento Móvel) - interage com o banco de dados local do equipamento móvel, executa a TM no banco de dados local e mantém a comunicação entre os ambientes móvel e de rede fixa, além de controlar a consistência local do banco de dados.
- Nível 3 (Estação de Suporte à Mobilidade) - representa a Transação Móvel (TM) na rede fixa, controla a ordem de execução das transações enviadas do ambiente móvel, controla a execução da TM na rede fixa, devolve o resultado do processamento da TM na rede fixa, acompanha a localização do cliente móvel e mantém a comunicação entre os ambientes móvel e de rede fixa;
- Nível 4 (Ambiente do SGBD) - Interage com o banco de dados local da rede fixa, executa as transações no banco de dados local e controla a consistência do banco de dados.

A transação-raiz e as transações-componentes desse modelo são, conceitualmente, assim definidas:

- **Transação Móvel (TM)** - é uma transação de nível 1, também chamada de *transação-raiz*, preparada e disparada pela aplicação móvel. Essa transação sempre gera duas transações-componentes de nível 2, as Transações Móveis Locais Compensáveis (TMLCs) e as Transações Móveis Globais (TMGs). As TMLCs serão executadas no próprio equipamento móvel, quando houver um SGBD local, e as TMGs serão sempre executadas no SGBD localizado na rede fixa. Quando possível, o gerente de transações locais do SGBD gera também as correspondentes Transações Móveis Locais Compensatórias (TMLCPs), para desfazer semanticamente o efeito das TMLCs canceladas após terem sido confirmadas (*commit*) no banco de dados local. Caso contrário, existe a necessidade de intervenção de um operador para gerar as TMLCPs.



- **Transação Móvel Local Compensável (TMLC)** - é uma transação de nível 2, gerada para ser executada no equipamento móvel, sempre que este possuir um banco de dados local. Essas *TMLCs*, quando realizarem atualizações, serão executadas de forma *serial*, sem concorrência, gerando bloqueios no banco de dados. Após ser confirmada (*commit*) no banco de dados local, só pode ter o seu efeito sobre os objetos do banco de dados cancelado, com a execução de uma Transação Móvel Local Compensatória (TMLCP) correspondente, caso haja divergência entre os esquemas dos bancos de dados.
- **Transação Móvel Global (TMG)** - é uma transação de nível 2, também chamada de *transação-raiz global*, que é responsável pela execução das operações da TM nos SGBDs localizados na rede fixa. A *transação-raiz global* executará todos os controles no equipamento móvel e só será confirmada (*commit*) quando todas as suas transações componentes também forem confirmadas (*commit*). Enquanto não receber a confirmação (*commit*) de todas as suas transações componentes oriundas da rede fixa, permanecerá ativa, e *nunca será cancelada*, mesmo que o equipamento móvel seja desconectado ou desligado.
- **Transação Móvel Local Compensatória (TMLCP)** - é uma transação de nível 2, gerada para ser executada no equipamento móvel sempre que uma Transação Móvel Local Compensável (TMLC) confirmada (*commit*), no banco de dados local, necessitar ter anulado o seu efeito sobre os objetos do banco de dados local por haver divergência entre os esquemas. Caracteriza-se por não poder falhar e por ser sempre executada com uma prioridade de execução superior a todas as transações TMLC, bloqueando todo o banco de dados local.
- **Transação Coordenadora de Mobilidade (TCM)** - é uma transação de nível 3, representante da Transação Móvel Global (TMG) na rede fixa. Será executada, preferencialmente, nas *Estações de Suporte à Mobilidade (ESB)* ou em algum equipamento da rede fixa e desempenhará, principalmente, o papel de coordenar a mobilidade dos clientes, garantindo assim que os resultados das transações processadas nos SGBDs, localizados na rede fixa, sejam entregues aos clientes móveis, mesmo que esses tenham se deslocado para a área de outra ESM. Assim como a Transação Móvel Global (TMG), só será confirmada (*commit*) quando todas as suas transações componentes também forem confirmadas (*commit*). Entretanto, quando isso acontecer, ficará em um estado de *temporariamente confirmada*, até

que a transação TMG assinale que foi confirmada (*commit*) com sucesso. Enquanto não receber essa confirmação final, permanecerá ativa e *nunca será cancelada*, mesmo que o equipamento em que esteja sendo executada seja desligado.

- **Transação Coordenadora de Banco de Dados (TCBD)** - transação de nível 3, é também uma representante da Transação Móvel Global (TMG) e da Transação Coordenadora de Mobilidade (TCM) na rede fixa. Será, preferencialmente, executada nas Estações de Suporte à Mobilidade (ESB), ou em outro equipamento da rede fixa, e somente desempenhará o papel de coordenadora da execução da Transação Móvel (TM) no banco de dados localizado na rede fixa. Possui todas as características de uma transação distribuída, porém sempre será a coordenadora do processamento das suas transações componentes, mesmo que haja qualquer tipo de falha no processamento. A consistência dos dados será garantida por meio da utilização do protocolo de confirmação em duas fases (*two phase commit - 2pc*). Essa transação só será confirmada (*commit*) quando todas as transações componentes também o forem (*commit*).
- **Transação de Suporte à Mobilidade do Cliente (TSMC)** - é uma transação de nível 3 que se caracteriza por ser somente de leitura (*read only*). Ela obtém a localização atual do cliente móvel, no banco de dados de localização, presente na estação de suporte à mobilidade, possibilitando assim a entrega dos resultados da Transação Móvel, processada nos SGBDs da rede fixa, ao cliente móvel. Uma vez identificadas a conexão e a localização do cliente móvel, também possibilita a interação entre a Transação Coordenadora de Mobilidade (TCM) e a Transação Móvel Global (TMG), indicando os momentos em que a interação deve ocorrer, informando sobre os *eventos relevantes* ocorridos no processamento das transações componentes.
- **Transação Local (TL)** - é uma transação de nível 4, que será executada e confirmada em um SGBD local, localizado na rede fixa, onde todos os requisitos e necessidades de um ambiente distribuído, por hipótese, já estão satisfeitos. Geralmente utilizando o protocolo de bloqueio dos dados em duas fases (*two phase locking*) ou um outro protocolo específico do SGBD local, será inserida em um escalonamento, o qual garantirá a consistência do banco de dados local.

As Transações Móveis (TM), Móvel Local Compensável (TMLC), Móvel Local Compensatória (TMLCP) e Local (TL) são definidas como

*uma abstração que representa a seqüência de operações de banco de dados resultante da execução de um programa de aplicação*, conforme enunciado no conceito clássico de transações na seção 2.2.1 do capítulo 2. Já as transações Móvel Global (TMG), Coordenadora de Mobilidade (TCM), Coordenadora de Banco de Dados (TCBD) e de Suporte à Mobilidade do Cliente (TSMC) possuem operações de controle que objetivam garantir a execução correta das demais transações sobre os bancos de dados localizados nos equipamentos móveis e da rede fixa.

#### 4.2.2

##### Os Estados de uma Transação Móvel no Modelo Proposto

No modelo proposto, uma *Transação Móvel (TM)* é definida como a raiz de uma árvore de transações, definida explicitamente na aplicação móvel, através das operações {Begin-Transaction-TM, [Abort-Transaction-TM, End-Transaction-TM]}. Essas operações delimitam o início e o fim de possíveis eventos sobre os objetos do banco de dados local (localizado no equipamento móvel) e global (localizado na rede fixa).

Essa transação é uma unidade atômica de trabalho que deverá ser completada integralmente no SGBD, localizado no equipamento móvel, e nos SGBDs, localizados na rede fixa, ou simplesmente não ser realizada. Para fins de recuperação em caso de falhas, o sistema de gerência de transações precisa acompanhar a transação, desde o momento em que a transação se inicia até quando termina, sendo confirmada ou cancelada. Portanto, as seguintes operações devem ser acompanhadas:

- **Begin-Transaction** - marca o início da execução da transação;
- **Read ou Write** - especifica operações de leitura e gravação que são executadas como parte da transação;
- **End-Transaction** - marca o final da execução da transação. Nesse ponto, é necessário verificar se as alterações introduzidas pela transação podem ser permanentemente aplicadas ao banco de dados (commit) ou se têm que ser canceladas (abort) porque violam a serialização ou por algum outro motivo;
- **Commit-Transaction** - sinaliza o final bem-sucedido da transação, de modo que as alterações introduzidas pela transação podem ser seguramente confirmadas (commit) no banco de dados e não poderão mais ser desfeitas;

- **Rollback ou Abort** - sinaliza que a transação terminou sem sucesso, de modo que as alterações introduzidas pela transação no banco de dados devem ser desfeitas;
- **Preparada-para-Commit** - indica que a transação está pronta para efetivar a confirmação (commit). A partir deste instante, o SGBD local não poderá cancelar a transação de forma unilateral. O SGBD local ficará aguardando a decisão do SGBD coordenador para emitir o comando de commit ou abort de sua transação local;
- **Requisição-de-serviço** - as transações globais podem requisitar a execução de funções providas pelos SGBDs locais, como por exemplo para persistência dos dados.

Conforme mostrado na figura 4.1, a *Transação Móvel (TM)* é composta pela *Transação Móvel Local Compensável (TMLC)* e pela *Transação Móvel Global (TMG)*. Assim, acompanhar as operação da Transação Móvel (TM) implica acompanhar as operações de suas duas transações componentes. A figura 4.2 apresenta um diagrama de transição de estados da Transação Móvel (TM), da Transação Móvel Local Compensável (TMLC) e da Transação Móvel Global (TMG) no modelo proposto, considerando que o resultado final da TM depende do resultado final de suas transações componentes.

### Estados da Transação Móvel (TM)

- **Ativa** - após iniciar a sua execução, a TM vai para o estado ativo. Nesse estado, é invocado um método de Criação de Transações para criar a Transação Móvel Local Compensável (TMLC) e a Transação Móvel Global (TMG). Ela ficará nesse estado até que a TMG tenha sido confirmada.
- **Parcialmente Confirmada** - neste ponto, todas as suas transações componentes, TMLC e TMG, já terminaram e confirmaram (*commit*) a sua execução. O resultado da sua execução é informado para a aplicação, e todas as Transações Móveis Locais Compensatórias (TMLCP) correspondentes são eliminadas.
- **Falhada** (*abort*) - neste ponto, pelo menos uma de suas transações componentes falhou. Caso tenha sido a TMLC, a TM cancelará a TMG. Caso tenha sido a TMG, a TM invocará a TMLCP para desfazer os efeitos de sua respectiva TMLC no banco de dados local.

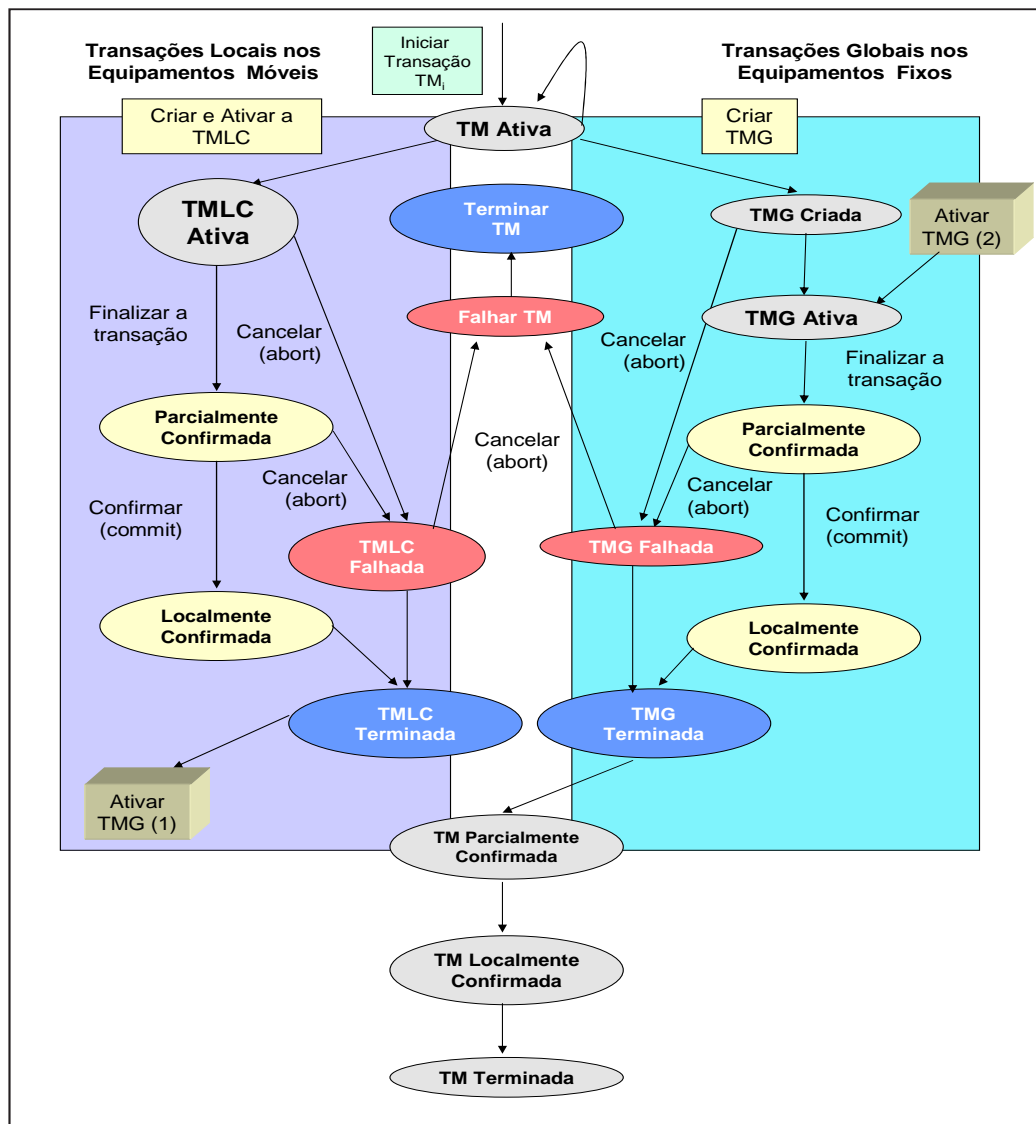


Figura 4.2: Diagrama de Transição de Estados de uma transação móvel (TM) no modelo proposto

- **Localmente Confirmada** - a TM escreve, no Histórico-M<sup>3</sup>, um registro informando o seu término.
- **Terminada** - a TM deixa o sistema, e todos os registros temporários sobre ela são eliminados dos catálogos de controle.

### Estados da Transação Móvel Local Compensável (TMLC)

- **Ativa** - após iniciar a sua execução, a TMLC vai para o estado ativo. Nesse estado, ela pode emitir todas as suas operações de leitura (*read*) e gravação (*write*) sobre os objetos do banco de dados. Se todas as

<sup>3</sup>Históricos são definidos no Anexo B.

suas operações foram realizadas com sucesso, ela passa para o estado de parcialmente confirmada. Caso haja alguma falha, ela vai para o estado de falhada.

- **Parcialmente Confirmada** - neste ponto, são executados alguns protocolos de segurança para garantir que, caso haja alguma falha no sistema, o subsistema de recuperação será capaz de garantir a confirmação (*commit*) da transação no banco de dados ou de desfazer os seus efeitos (*rollback*).
- **Falhada** - todas as operações de gravação da transação terão que ser desfeitas (*rollback*) para tornar o banco de dados novamente consistente.
- **Localmente Confirmada** - a TMLC conclui a sua execução com sucesso e todas as suas alterações são registradas de forma permanente no banco de dados.
- **Terminada** - a TMLC ativa a TMG correspondente à sua TM e deixa o sistema. Todos os seus registros temporários são eliminados dos catálogos de controle do sistema.

### Estados da Transação Móvel Global (TMG)

- **Ativa** - após o término com sucesso da TMLC, a TMG inicia a sua execução e vai para o estado ativo. Nesse estado, ela gera a Transação Coordenadora de Mobilidade (TCM), que é enviada para execução na rede fixa. Ela permanecerá nesse estado até que receba o resultado da TCM.
- **Parcialmente Confirmada** - neste ponto, todas as transações componentes de sua árvore de execução, que foram executadas na rede fixa, já terminaram a sua execução e foram confirmadas (*commit*). A TMG informa à TM o resultado da sua execução e se prepara para terminar a sua execução.
- **Falhada** - neste ponto, pelo menos uma de suas transações componentes, que executou na rede fixa, falhou, levando, conseqüentemente, à falha da TCM. A TMG avisa à TM que não terminará com sucesso.
- **Localmente Confirmada** - a TMG escreve, no Histórico-M, um registro informando o seu término e avisa à TCM que será confirmada com sucesso.
- **Terminada** - a TMG deixa o sistema e todos os registros temporários sobre ela são eliminados dos catálogos de controle.

### 4.3

#### Formalização do Modelo de Transações Proposto

Nesta seção, é apresentada a semântica do modelo de transações proposto. São definidas as estruturas e a semântica operacional de cada um dos tipos de transação. As definições apresentadas seguem o padrão encontrado em [9] e o formalismo ACTA definido em [12], [13] e [14], descrito no Anexo B.

Para o **ambiente da computação móvel**, dois históricos são considerados:

1. *Histórico do Banco de Dados Móvel (Histórico-M)* - esse histórico registra todos os *eventos significativos* e os *eventos de objetos*<sup>4</sup> que ocorrem nos bancos de dados localizados nos equipamentos móveis, em função das *Transações Móveis (TMs)*, das *Transações Móveis Locais Compensáveis (TMLCs)*, das *Transações Móveis Globais (TMGs)* e das *Transações Móveis Locais Compensatórias (TMLCPs)*. Entretanto, para as TMGs, são registrados apenas os eventos significativos, com o objetivo de capturar as dependências entre as transações;
2. *Histórico do Banco de Dados localizado na Rede Fixa (Histórico-F)* - esse histórico registra todos os *eventos significativos* e os *eventos de objetos* que ocorrem nos bancos de dados localizados nos equipamentos da rede fixa, em função das demais transações, acrescidas das *Transações Móveis Globais (TMGs)*.

#### 4.3.1

##### A Transação Móvel (TM)

Uma *Transação Móvel (TM)* é a raiz de uma árvore de transações, que é definida explicitamente na *aplicação móvel*, através das operações {Begin-Transaction-TM, [Abort-Transaction-TM, End-Transaction-TM]}. Essas operações delimitam o início e o fim de possíveis eventos sobre os objetos do banco de dados local (localizado no equipamento móvel) e global (localizado na rede fixa).

**Definição 1:** Uma Transação Móvel  $TM_i$  é composta por um conjunto  $CT_i$  de transações, o qual guarda uma relação de ordenação  $\angle_i$ , tal que:

---

<sup>4</sup>Eventos significativos e eventos de objetos estão definidos na seção B.2 do Anexo B.

1.  $TM_i = \{\text{Begin-Transaction-}TM_i\} \cup \{CT_i\} \cup \{\text{Abort-transaction-}TM_i, \text{End-Transaction-}TM_i\}$
2.  $CT_i = (\{TMLC_i\} \cup \{TMG_i\} \cup \{TMLCP_i\}) \vee (\{TMG_i\}, \text{ se } \bar{A} \text{ BD local})$
3.  $\forall t \in CT_i (\text{Begin-Transaction-}TM_i \angle_i t) \wedge (t \angle_i \{\text{Abort-Transaction-}TM_i, \text{End-Transaction-}TM_i\})$
4.  $\forall TMLC_i, TMLCP_i \in CT_i (TMLC_i \angle_i TMLCP_i)$
5.  $\forall TMLC_i, TMG_i \in CT_i (TMLC_i \angle_i TMG_i)$
6.  $\forall TMG_i, TMLCP_i \in CT_i (TMG_i \angle_i TMLCP_i)$

O *item 1* da definição 1 apresenta a composição da transação  $TM_i$  como sendo um conjunto de três transações aninhadas, precedidas pelo *evento* de início da *transação móvel* e seguidas pelo *evento* de término da *transação móvel*.

O *item 2* da definição 1 apresenta as transações componentes da transação móvel  $TM_i$ , conforme a definição já apresentada para as transações  $TMLC_i$ ,  $TMG_i$  e  $TMLCP_i$ . Entretanto, caso o banco de dados não esteja replicado no equipamento móvel, a transação móvel será composta apenas pela Transação Móvel Global (TM) e o *Histórico-M* conterà apenas os *eventos significativos* das Transações Móveis Globais.

O *item 3* da definição 1 define a condição de término da transação móvel  $TM_i$  e estabelece que, após ser concluída, ela não pode ser sucedida por qualquer outra transação componente.

Os *itens de 4 a 6* da definição 1 estabelecem a ordem e as precedências entre os tipos de transações nas seqüências de execução. Assim, estabelecem-se as seguintes dependências para execução das transações nos equipamentos móveis:

#### **Dependência Forte de Confirmação para TM ( $TMG_i$ SCD $TM_i$ )**

- $(\text{End-Transaction-}TM_i \in \text{Histórico-M}) \Rightarrow ((\text{Commit}_{TMG_i} \in \text{Histórico-M}) \Rightarrow (\text{Commit}_{TMG_i} \rightarrow \text{End-Transaction-}TM_i))$ , ou
- $(\text{Commit}_{TMG_i} \in \text{Histórico-M}) \Rightarrow (\text{Commit}_{TM_i} \in \text{Histórico-M})$

Essa dependência significa que a transação móvel  $TM_i$  só pode ter a sua execução confirmada (*commit*) se a sua Transação Móvel Global  $TMG_i$  também tiver a execução confirmada (*commit*).



**Dependência de Cancelamento para TM ( $TMG_i$  AD  $TM_i$ )**

- ( $Abort_{TMG_i} \in \text{Histórico-M}$ )  $\Rightarrow$  ( $Abort_{TM_i} \in \text{Histórico-M}$ )

Essa dependência significa que o cancelamento (*abort*) da Transação Móvel Global  $TMG_i$  implica o cancelamento da transação móvel  $TM_i$ .

**Dependência de Término para TM ( $TMG_i$  TD  $TM_i$ )**

- ( $Commit_{TMG_i} \in \text{Histórico-M}$ )  $\Rightarrow$  ( $\epsilon \rightarrow \epsilon'$ ) onde  
 $\epsilon \in \{Commit_{TM_i}, Abort_{TM_i}\}$  e  
 $\epsilon' \in \{Commit_{TMG_i}, Abort_{TMG_i}\}$

Essa dependência significa que a Transação Móvel  $TM_i$  não pode ser confirmada (*commit*) ou cancelada (*abort*) até que a Transação Móvel Global  $TMG_i$  seja confirmada ou cancelada.

**Dependência Serial para TM ( $TM_j$  SD  $TM_i$ )**

- ( $Begin_{TM_j} \in \text{Histórico-M}$ )  $\Rightarrow$  ( $\epsilon \rightarrow Begin_{TM_j}$ ) onde  
 $\epsilon \in \{Commit_{TM_i}, Abort_{TM_i}\}$

Essa dependência significa que as Transações Móveis (TMs) serão executadas de forma *serial* no banco de dados local e no banco de dados localizado na rede fixa.

As *Transações Móveis* não possuem as dependências de exclusão, de confirmação forçada no cancelamento, de início, de início de confirmação e de início de cancelamento.

**Regra de Visibilidade<sup>5</sup> para a TM**

- $ViewSet_{TM_i} = \emptyset$

Essa regra significa que a Transação Móvel  $TM_i$  só opera sobre o banco de dados através de suas transações componentes, ou seja, via as transações  $TMLC_i$ ,  $TMG_i$  e  $TMLCP_i$ .

---

<sup>5</sup>A Regra de Visibilidade está definida no Anexo B.

### 4.3.2

#### A Transação Móvel Local Compensável (TMLC)

Uma *Transação Móvel Local Compensável (TMLC)* é uma transação componente da *Transação Móvel (TM)* projetada para executar no banco de dados local residente no equipamento móvel. Ela possui um conjunto de operações sobre os objetos do banco de dados local. É definida explicitamente pela aplicação móvel através dos comandos {Begin-Transaction-TMLC, {Commit-Transaction-TMLC, Abort-Transaction-TMLC}}, delimitadores do escopo da transação.

**Definição 2:** Sejam  $TM_i$  e  $TM_j$  duas Transações Móveis,  $E_l$  um esquema local de um banco de dados, C-Obj nomes de conjuntos de objetos, C-Proc nomes de conjuntos de procedimentos, onde  $(C-Obj, C-Proc) \in E_l$ . Uma Transação Móvel Local Compensável  $TMLC_i \in TM_i$  é composta por um conjunto de operações sobre os objetos do banco de dados e é uma ordem parcial com relação de ordenação  $\angle_i$ , onde:

1.  $TMLC_i = \{\text{Begin-Transaction-}TMLC_i\} \cup \{p_{ik}[\text{obj}] \mid p_{ik} \in \text{C-Proc}, \text{obj} \in \text{C-Obj}\} \cup \{\text{Abort-Transaction-}TMLC_i, \text{Commit-Transaction-}TMLC_i\}$
2. Para quaisquer duas solicitações de procedimentos  $(p_{ik}[\text{obj}], p_{jl}[\text{obj}]) \in \text{C-Proc}$ , se **conflitam**<sup>6</sup>  $(\langle \text{obj}, p_{ik} \rangle, \langle \text{obj}, p_{jl} \rangle)$ , então  $(p_{ik}[\text{obj}] \angle_i p_{jl}[\text{obj}]) \vee (p_{jl}[\text{obj}] \angle_i p_{ik}[\text{obj}])$
3.  $\forall p_{ik} \in \text{C-Proc} ((\{\text{Begin-Transaction-}TMLC_i\} \angle_i p_{ik}) \wedge (p_{ik} \angle_i \{\text{Abort-Transaction-}TMLC_i, \text{Commit-Transaction-}TMLC_i\}))$

O *item 1* da definição 2 determina os delimitadores de início e término da Transação Móvel Local Compensável (TMLC), bem como o conjunto de procedimentos (métodos) que podem ser executados sobre os objetos do banco de dados.

O *item 2* da definição 2 especifica a ordem em que as operações conflitantes devem ser executadas.

O *item 3* da definição 2 especifica que, após o evento de término da transação, nenhum outro procedimento pode ser executado, ou seja, ele não é sucedido por qualquer procedimento.

A seguir estão definidas as dependências para execução das transações nos equipamentos móveis:

<sup>6</sup>O conceito de conflito entre transações está definido na seção 2.2.2, *Definição 4*, do capítulo 2.

**Dependência de Cancelamento para a TMLC ( $TMLC_i$  AD  $TM_i$ )**

- (Abort- $TM_i \in$  Histórico-M)  $\Rightarrow$  (Abort- $TMLC_i \in$  Histórico-M)  $\vee$   
 ((Commit- $TMLC_i \in$  Histórico-M)  $\wedge$  (Commit- $TMLCP_i \in$   
 Histórico-M))

Essa dependência, que na verdade é um *axioma* do modelo proposto, significa que, se a *Transação Móvel*  $TM_i$  for cancelada, a transação  $TMLC_i$  também será cancelada. No caso de o cancelamento da *Transação Móvel*  $TM_i$  ocorrer após a confirmação da transação  $TMLC_i$ , essa transação será compensada no banco de dados, através de uma *Transação Móvel Local Compensatória*  $TMLCP_i$ .

**Dependência Serial para a TMLC ( $TMLC_j$  SD  $TMLC_i$ )**

- (Begin- $TMLC_j \in$  Histórico-M)  $\Rightarrow$  ( $\varepsilon \rightarrow$  Begin- $TMLC_j$ ) onde  
 $\varepsilon \in \{\text{Commit-}TMLC_i, \text{Abort-}TMLC_i\}$

Essa dependência significa que uma *transação móvel local compensável* não pode iniciar a sua execução até que a transação que a precede tenha sido confirmada (*commit*) ou cancelada (*abort*).

As *Transações Móveis Locais Compensáveis* não possuem as dependências de confirmação, de término, de exclusão, de confirmação forçada no cancelamento, de início, de início de confirmação e de início de cancelamento.

**Regra de Visibilidade para a TMLC**

- $\text{ViewSet}_{TMLC_i} = \{\text{BD-Móvel}_i\}$

Essa regra significa que a *Transação Móvel Local Compensável*  $TMLC_i$  opera diretamente sobre os objetos do banco de dados, através de operações definidas no seu esquema local, visualizando o último estado confirmado (*commit*) do banco de dados.

### 4.3.3

#### A Transação Móvel Global (TMG)

Uma *Transação Móvel Global (TMG)* é uma subárvore da árvore de transações da *Transação Móvel (TM)*, definida explicitamente na *aplicação móvel* por meio das operações {Begin-Transaction-TMG, [Abort-Transaction-TMG, End-Transaction-TMG]}. Essas operações delimitam o início e o fim de possíveis eventos sobre os objetos do banco de dados global localizado na rede fixa.

**Definição 3:** Uma Transação Móvel Global  $TMG_i$  é composta por uma única *Transação Coordenadora de Mobilidade  $TCM_i$* , que será sua representante na rede fixa.

1.  $TMG_i = \{\text{Begin-Transaction-}TMG_i\} \cup \{TCM_i\} \cup \{\text{Abort-Transaction-}TMG_i, \text{End-Transaction-}TMG_i\}$
2.  $\forall tcm \in TCM_i (\text{Begin-Transaction-}TMG_i \angle_i tcm) \wedge (tcm \angle_i \{\text{Abort-Transaction-}TMG_i, \text{End-Transaction-}TMG_i\})$

O *item 1* da definição 3 apresenta a composição da transação  $TMG_i$  como uma TCM, precedida pelo *evento* de início da Transação Móvel Global e seguida pelo *evento* de término da Transação Móvel Global.

O *item 2* da definição 3 estabelece a condição de término da Transação Móvel Global  $TMG_i$  e determina que, após ser concluída, ela não pode ser sucedida por qualquer outra transação componente.

#### Dependência de Confirmação da TMG ( $TMG_i$ CD $TMLC_i$ )

- $(\text{Commit}_{TMG_i} \in \text{Histórico-M}) \Rightarrow ((\text{Commit}_{TMLC_i} \in \text{Histórico-M}) \Rightarrow (\text{Commit}_{TMLC_i} \rightarrow \text{Commit}_{TMG_i}))$

Essa dependência significa que, se a Transação Móvel Local Compensável  $TMLC_i$  e a Transação Móvel Global  $TMG_i$  são confirmadas (*commit*), então a confirmação da transação  $TMLC_i$  precede a confirmação da transação  $TMG_i$ .

#### Dependência Forte de Confirmação para TMG ( $TCM_i$ SCD $TMG_i$ )

- $(\text{End-Transaction-}TMG_i \in \text{Histórico-M}) \Rightarrow (((\text{Commit}_{TCM_i} \in \text{Histórico-F}) \Rightarrow ((\text{Commit}_{TCM_i}) \rightarrow \text{End-Transaction-}TMG_i)), \text{ou})$
- $(\text{Commit}_{TCM_i} \in \text{Histórico-F}) \Rightarrow (\text{Commit}_{TMG_i} \in \text{Histórico-M})$

Essa dependência significa que a Transação Móvel Global  $TMG_i$  só pode ter a sua execução confirmada (*commit*) se a sua transação componente  $TCM_i$  também tiver a execução confirmada (*commit*). Observe que, nesse caso, o modelo exige o acompanhamento dos *históricos das transações*, armazenados nos SGBDs localizados na rede fixa e no equipamento móvel.

#### Dependência de Cancelamento para TMG ( $TCM_i$ AD $TMG_i$ )

- ( $Abort_{TCM_i} \in \text{Histórico-F}$ )  $\Rightarrow$  ( $Abort_{TMG_i} \in \text{Histórico-M}$ )

Essa dependência significa que o cancelamento (*abort*) da Transação Coordenadora de Mobilidade  $TCM_i$  implica o cancelamento da Transação Móvel Global  $TMG_i$ .

#### Dependência de Término para TMG ( $TCM_i$ SD $TMG_i$ )

- ( $(Commit_{TCM_i} \in \text{Histórico-F}) \Rightarrow (\epsilon \rightarrow \epsilon')$  onde  
 $\epsilon \in \{Commit_{TMG_i}\}$  e  
 $\epsilon' \in (\{Commit_{TCM_i}\})$ )
- ( $(Abort_{TCM_i} \in \text{Histórico-F}) \Rightarrow (\epsilon \rightarrow \epsilon')$  onde  
 $\epsilon \in \{Abort_{TMG_i}\}$  e  
 $\epsilon' \in (\{Abort_{TCM_i}\})$ )

Essa dependência significa que a *Transação Móvel Global*  $TMG_i$  não pode ser confirmada (*commit*) e registrada no Histórico-M até que a transação  $TCM_i$  seja confirmada (*commit*) e registrada no Histórico-F. Da mesma forma, ela somente pode ser cancelada (*abort*) e registrada no Histórico-M se a transação  $TCM_i$  for cancelada (*abort*) e registrada no Histórico-F.

#### Dependência Fraca de Início na Confirmação da TMG ( $TMG_i$ WCD $TMLC_i$ )

- ( $Begin_{TMLC_i} \in \text{Histórico-M}$ )  $\Rightarrow$  ( $(Commit_{TMG_i} \in \text{Histórico-M}) \Rightarrow$   
 $(Commit_{TMLC_i} \rightarrow Begin_{TMG_i}))$

Essa dependência significa que, se a *Transação Móvel Local Compensável*  $TMLC_i$  é confirmada e registrada no Histórico-F, a Transação Móvel Global  $TMG_i$  pode iniciar a sua execução logo após a confirmação de  $TMLC_i$ . *Ou seja, a Transação Móvel Global só será executada se a sua transação local for confirmada.* No caso de não existir um banco de dados local no equipamento móvel, essa dependência não existe.

As *Transações Móveis Globais* não possuem as dependências de exclusão, de confirmação forçada no cancelamento, de início, serial, de início e de início de cancelamento.

## Regra de Visibilidade para a TMG

$$- \text{ViewSet}_{TMG_i} = \emptyset$$

Essa regra significa que a Transação Móvel Global  $TM_i$  só opera sobre o banco de dados por meio de suas transações componentes, ou seja, via as transações  $TCBDC_i$  e  $TCM_i$ .

### 4.3.4

#### A Transação Móvel Local Compensatória (TMLCP)

Uma *Transação Móvel Local Compensatória (TMLCP)* é composta por um conjunto de operações sobre os objetos do banco de dados local e objetiva desfazer semanticamente (*compensar*) os efeitos da *Transação Móvel Local Compensável (TMLC)*, executada e confirmada no banco de dados local, localizado no equipamento móvel. Essa transação pode ser construída pelo sistema de gerência de transações, a partir das invocações realizadas pela TMLC, por intermédio de métodos definidos como compensáveis, e pelas definições dos respectivos conjuntos de funções compensatórias, também já definidos ou por intervenção de um operador autorizado. Durante a execução da TMLC, o sistema de gerência de transações compõe a TMLCP correspondente. Quando a TMLC é aceita, a sua TMLCP correspondente é "empilhada". Caso a Transação Móvel (TM) seja aceita, a TMLCP é descartada, e, no caso de a *Transação Móvel (TM)* ser cancelada, a TMLCP é então executada. Uma TMLCP é definida explicitamente pelos comandos  $\{\text{Begin-Transaction-TMLCP}\}$ ,  $\{\text{Commit-Transaction-TMLCP}\}$ ,  $\{\text{Abort-Transaction-TMLCP}\}$ , delimitadores do escopo da transação.

**Definição 4:** Sejam  $TM_i$  e  $TM_j$  duas *Transações Móveis*,  $E_l$  um esquema local de um banco de dados localizado no equipamento móvel, C-Obj nomes de conjuntos de objetos, C-Proc nomes de conjuntos de procedimentos, onde  $(\text{C-Obj}, \text{C-Proc}) \in E_l$ . Uma *Transação Móvel Local Compensatória*  $TMLCP_i \in TM_i$  é uma ordem parcial com relação de ordenação  $\angle_i$ , onde:

1.  $TMLCP_i = \{\text{Begin-Transaction-TMLCP}_i\} \cup \{p_{ik}[\text{obj}] \mid p_{ik} \in \text{C-Proc}, \text{obj} \in \text{C-Obj}\} \cup \{\text{Abort-Transaction-TMLCP}_i, \text{Commit-Transaction-TMLCP}_i\}$
2. Para qualquer duas solicitações de procedimentos  $(p_{ik}[\text{obj}], p_{jl}[\text{obj}]) \in \text{C-Proc}$ , se conflitam  $(\langle \text{obj}, p_{ik} \rangle, \langle \text{obj}, p_{jl} \rangle)$ , então  $(p_{ik}[\text{obj}] \angle_i p_{jl}[\text{obj}]) \vee (p_{jl}[\text{obj}] \angle_i p_{ik}[\text{obj}])$

3.  $\forall p_{ik} \in \text{C-Proc} ((\{\text{Begin-Transaction-}TMLCP_i\} \angle_i p_{ik}) \wedge (p_{ik} \angle_i \{\text{Abort-Transaction-}TMLCP_i, \text{Commit-Transaction-}TMLCP_i\}))$
4.  $\forall TMLCP_i \in TM_i (TMLC_i \angle_i TMLCP_i)$

O *item 1* da definição 4 estabelece os delimitadores de início e término da *Transação Móvel Local Compensatória (TMLCP)*, bem como o conjunto de procedimentos (métodos) que podem ser executados sobre os objetos do banco de dados.

É importante ressaltar que a TMLCP não pode falhar, embora haja a possibilidade de existir ( $\text{Abort-Transaction-}TMLCP_i \in \text{Histórico-M}$ ) no caso de a TMLCP ter sido cancelada pelo SGBD local por algum critério unilateral.

O *item 2* da definição 4 especifica a ordem em que as operações conflitantes devem ser executadas.

O *item 3* da definição 4 especifica que, após o evento de término da transação, nenhum outro procedimento pode ser executado, ou seja, ele não é sucedido por qualquer procedimento.

O *item 4* da definição 4 especifica a ordem e as precedências entre as transações TMLC e TMLCP, nas seqüências de execução.

A seguir, estão definidas as dependências para execução das TMLCP nos equipamentos móveis.

#### **Dependência de Início na Confirmação para a TMLCP ( $TMLCP_i$ BD $TMLC_i$ )**

- ( $\text{Begin-Transaction-}TMLCP_i \in \text{Histórico-M}$ )  $\Rightarrow$  ( $\text{Commit}_{TMLC} \rightarrow \text{Begin-}TMLCP_i$ )

Essa dependência significa que uma *Transação Móvel Local Compensatória TMLCP<sub>i</sub>* só pode iniciar a sua execução se a sua correspondente *Transação Móvel Local Compensável TMLC<sub>i</sub>* tiver sido executada e confirmada (*commit*) com sucesso no banco de dados local.

- ( $\forall TMLCP_i \in TM_i (\text{Commit}_{TMG} \in \text{Histórico-M}) \Rightarrow (\text{Begin-Transaction-}TMLCP_i \notin \text{Histórico-M})$ )
- $TMLCP_i$  **BD**  $TMG_i$

Essa dependência significa que uma *Transação Móvel Local Compensatória (TMLCP<sub>i</sub>)* não pode ser executada se a *Transação Móvel Global (TMG<sub>i</sub>)* já tiver sido confirmada (*commit*). A confirmação da TMG leva à confirmação imediata da *Transação Móvel (TM)*. Essa condição evita

que uma transação TMLCP desfaza o resultado da execução da transação TMLC executada no equipamento móvel, após a execução e confirmação da TM, no SGBD localizado na rede fixa.

### Dependência de Início de Cancelamento para a TMLCP

$(TMLCP_i \text{ BAD } TMG_i)$

- (Begin-Transaction- $TMLCP_i \in$  Histórico-M)  $\Rightarrow$   
(Abort-Transaction- $TMG_i \rightarrow$  Begin-Transaction- $TMLCP_i$ )

Essa dependência significa que o cancelamento da *Transação Móvel Global*  $TMG_i$  e, por conseguinte, da *Transação Móvel*  $TM_i$  devem iniciar imediatamente a execução da  $TMLCP_i$ , no equipamento móvel, para desfazer os efeitos da  $TMLC_i$  no banco de dados local.

As *Transações Móveis Locais Compensatórias* não possuem as dependências de confirmação, de cancelamento, de término, de exclusão, de confirmação forçada no cancelamento, de início e serial.

### Regra de Visibilidade para a TMLCP

- $ViewSet_{TMLCP_i} = \{BD_i\}$

Essa regra significa que a *Transação Móvel Local Compensatória*  $TMLCP_i$  opera diretamente sobre os objetos do banco de dados, por meio de operações definidas no seu esquema local, visualizando o último estado confirmado (*commit*) do banco de dados.

#### 4.3.5

#### A Transação Coordenadora de Mobilidade (TCM)

Uma *Transação Coordenadora de Mobilidade (TCM)* é uma subárvore da árvore da *Transação Móvel (TM)*, e a representante da *Transação Móvel Global (TMG)* na rede fixa. É definida explicitamente através das operações {Begin-Transaction-TCM, [Abort-Transaction-TCM, End-Transaction-TCM]}. Essas operações delimitam o início e o fim de possíveis eventos sobre os objetos do banco de dados global, localizado na rede fixa. Por ocasião de sua definição, além das operações solicitadas pelo projetista da aplicação móvel, serão introduzidos, em seu código, procedimentos para localizar, no BD de localização dos clientes móveis, residente na Estação de Suporte à Mobilidade (ESB), a posição atual do cliente móvel.



**Definição 5:** Uma Transação Coordenadora de Mobilidade  $TCM_i$  é composta por um conjunto  $CT_i$  de transações que guarda uma relação de ordenação  $\angle_i$  tal que:

1.  $TCM_i = \{\text{Begin-Transaction-}TCM_i\} \cup \{CT_i\} \cup \{\text{Abort-Transaction-}TCM_i, \text{End-Transaction-}TCM_i\}$
2.  $CT_i = \{TCBD_i\} \cup \{TSMC_i\}$
3.  $\forall t \in CT_i (\text{Begin-Transaction-}TCM_i \angle_i t) \wedge (t \angle_i \{\text{Abort-Transaction-}TCM_i, \text{End-Transaction-}TCM_i\})$
4.  $\forall TCBD_i, TSMC_i \in CT_i (TCBD_i \angle_i TSMC_i)$

O *item 1* da definição 5 apresenta a composição da transação  $TCM_i$  como um conjunto de duas transações aninhadas, precedidas pelo *evento* de início da transação e seguido pelo *evento* de término da Transação Coordenadora de Mobilidade  $TCM_i$ .

O *item 2* da definição 5 apresenta as transações componentes da Transação Coordenadora de Mobilidade  $TCM_i$ . A TCM é composta pela Transação Coordenadora de Banco de Dados  $TCBD_i$  e pela Transação de Suporte à Mobilidade do Cliente  $TSMC_i$ .

O *item 3* da definição 5 determina a condição de término da Transação Coordenadora de Mobilidade  $TCM_i$  e estabelece que, após ser concluída, ela não pode ser sucedida por qualquer outra transação componente.

O *item 4* da definição 5 estabelece a ordem e a precedência entre os tipos de transações nas seqüências de execução. Assim, estabelecem-se as seguintes dependências para execução das transações na *rede fixa*:

#### **Dependência Forte de Confirmação para TCM ( $TCBD_i$ SCD $TCM_i$ ) $\wedge$ ( $TSMC_i$ SCD $TCM_i$ )**

- $(\text{End-Transaction-}TCM_i \in \text{Histórico-F}) \Rightarrow (((\text{Commit}_{TCBD_i} \in \text{Histórico-F}) \wedge (\text{Commit}_{TSMC_i} \in \text{Histórico-F}) \Rightarrow ((\text{Commit}_{TCBD_i} \wedge \text{Commit}_{TSMC_i}) \rightarrow \text{End-Transaction-}TCM_i)), \text{ou}$
- $((\text{Commit}_{TCBD_i} \in \text{Histórico-F}) \wedge (\text{Commit}_{TSMC_i} \in \text{Histórico-F})) \Rightarrow (\text{Commit}_{TCM_i} \in \text{Histórico-F})$

Essa dependência significa que a Transação Coordenadora de Mobilidade  $TCM_i$  só pode ter a sua execução confirmada (*commit*) se as suas transações componentes  $TCBD_i$  e  $TSMC_i$  também tiverem as execuções

confirmadas (*commit*). Observe que, nesse caso, o modelo exige o acompanhamento dos *históricos das transações*, armazenados nos SGBDs localizados na rede fixa.

### Dependência de Cancelamento para TCM ( $TCBD_i \text{ AD } TCM_i$ ) $\wedge$ ( $TSMC_i \text{ AD } TCM_i$ )

- $((\text{Abort}_{TCBD_i} \in \text{Histórico-F}) \vee (\text{Abort}_{TSMC_i} \in \text{Histórico-F})) \Rightarrow (\text{Abort}_{TCM_i} \in \text{Histórico-M})$

Essa dependência significa que o cancelamento (*abort*) da Transação Coordenadora de Banco de Dados  $TCBD_i$  ou da Transação de Suporte à Mobilidade do Cliente  $TSMC_i$  implica o cancelamento da Transação Coordenadora de Mobilidade  $TCM_i$ .

### Dependência de Término para TCM ( $TCBD_i \text{ SD } TCM_i$ ) $\wedge$ ( $TSMC_i \text{ SD } TCM_i$ )

- $((\text{Commit}_{TCBD_i} \in \text{Histórico-F}) \wedge (\text{Commit}_{TSMC_i} \in \text{Histórico-F})) \Rightarrow (\epsilon \rightarrow \epsilon')$  onde  
 $\epsilon \in \{\text{Commit}_{TCM_i}\}$  e  
 $\epsilon' \in (\{\text{Commit}_{TCBD_i}\} \wedge \{\text{Commit}_{TSMC_i}\})$
- $((\text{Abort}_{TCBD_i} \in \text{Histórico-F}) \wedge (\text{Abort}_{TSMC_i} \in \text{Histórico-F})) \Rightarrow (\epsilon \rightarrow \epsilon')$  onde  
 $\epsilon \in \{\text{Abort}_{TCM_i}\}$  e  
 $\epsilon' \in (\{\text{Abort}_{TCBD_i}\} \wedge \{\text{Abort}_{TSMC_i}\})$

Essa dependência significa que a *Transação Coordenadora de Mobilidade*  $TCM_i$  não pode ser confirmada (*commit*) e registrada no Histórico-M até que as transações  $TCBD_i$  e  $TSMC_i$  sejam confirmadas (*commit*) e registradas no Histórico-F. Da mesma forma, ela somente pode ser cancelada (*abort*) e registrada no Histórico-M se as transações  $TCBD_i$  e  $TSMC_i$  forem canceladas (*abort*) e registradas no Histórico-F.

As *Transações Coordenadoras de Mobilidade* não possuem as dependências de exclusão, de confirmação forçada no cancelamento, de início, serial, de início de confirmação e de início de cancelamento.

### Regra de Visibilidade para a TCM

- $\text{ViewSet}_{TCM_i} = \emptyset$

Essa regra significa que a Transação Coordenadora de Mobilidade  $TCM_i$  só opera sobre o banco de dados através de suas transações componentes, ou seja, via as transações  $TCBDC_i$  e  $TSMC_i$ .

### 4.3.6

#### Transação Coordenadora de Banco de Dados (TCBD)

Uma *Transação Coordenadora de Banco de Dados (TCBD)* é uma subárvore da *Transação Móvel (TM)* que é definida explicitamente na *aplicação móvel* através das operações {Begin-Transaction-TCBD, [Abort-Transaction-TCBD, End-Transaction-TCBD]}. Essas operações delimitam o início e o fim de possíveis eventos sobre os objetos dos bancos de dados, localizados na rede fixa.

**Definição 6:** Uma *Transação Coordenadora de Banco de Dados*  $TCBD_i$  é composta por um conjunto  $CT_i$  de *Transações Locais*  $TL_{in}$  que guarda uma relação de ordenação  $\angle_i$  tal que:

1.  $TCBD_i = \{\text{Begin-Transaction-}TCBD_i\} \cup \{CT_i\} \cup \{\text{Abort-transaction-}TCBD_i, \text{End-Transaction-}TCBD_i\}$
2.  $CT_i = \{TC_{i1}, TC_{i2}, TC_{i3}, \dots, TC_{in-2}, TC_{in-1}, TC_{in}\}$
3.  $\forall t \in CT_i (\text{Begin-Transaction-}TCBD_i \angle_i t) \wedge (t \angle_i \{\text{Abort-Transaction-}TCBD_i, \text{End-Transaction-}TCBD_i\})$

O *item 1* da definição 6 apresenta a composição da Transação Coordenadora de Banco de Dados  $TCBD_i$  como um conjunto de transações aninhadas, precedidas pelo *evento* de início da *transação móvel* e seguido pelo *evento* de término da *transação móvel*.

O *item 2* da definição 6 apresenta as transações componentes da Transação Coordenadora de Banco de Dados  $TCBD_i$ , conforme a definição já apresentada para as Transações Locais (TL).

O *item 3* da definição 6 determina a condição de término da Transação Coordenadora de Banco de Dados  $TCBD_i$  e estabelece que, após ser concluída, ela não pode ser sucedida por qualquer outra transação componente. Assim, estabelecem-se as seguintes dependências para execução das transações na rede fixa:

#### Dependência Forte de Confirmação para a TCBD ( $TL_i$ SCD $TCBD_i$ )

- $(\text{End-Transaction-}TCBD_i \in \text{Histórico-F}) \Rightarrow (\text{Commit}_{TL_{ik}}, \forall k \in \text{Histórico-F} \rightarrow \text{End-Transaction-}TCBD_i), \text{ ou}$
- $(\text{Commit}_{TL_{ik}}, \forall k \in \text{Histórico-F}) \Rightarrow (\text{Commit}_{TCBD_i} \in \text{Histórico-F})$

Essa dependência significa que a *Transação Coordenadora de Banco de Dados TCBD<sub>i</sub>* só pode ter a sua execução confirmada (*commit*) se todas as suas *Transações Locais TL<sub>in</sub>* também tiverem as execuções confirmadas (*commit*).

#### Dependência de Cancelamento para a TCBD ( $TL_i$ AD $TCBD_i$ )

- ( $Abort_{TL_{ik}}, (1 \leq k \leq n) \in \text{Histórico-F}$ )  $\Rightarrow$  ( $Abort_{TCBD_i} \in \text{Histórico-F}$ )

Essa dependência significa que o cancelamento (*abort*) de pelo menos uma *Transação Local TL<sub>i</sub>* implica o cancelamento da *Transação Coordenadora de Banco de Dados TCBD<sub>i</sub>*.

As *Transações Coordenadoras de Banco de Dados* não possuem as dependências de término, de exclusão, de confirmação forçada no cancelamento, de início, serial, de início de confirmação e de início de cancelamento.

#### Regra de Visibilidade para a TCBD

- $\text{ViewSet}_{CT_i} = \emptyset$

Essa regra significa que a *Transação Coordenadora de Banco de Dados TCBD<sub>i</sub>* só opera sobre o banco de dados, através de suas transações componentes, ou seja, via as transações  $TL_i$ .

### 4.3.7

#### Transação de Suporte à Mobilidade do Cliente (TSMC)

Uma *Transação de Suporte à Mobilidade do Cliente (TSMC)* é uma transação composta somente por operações de leitura sobre os objetos do banco de dados de localização dos clientes móveis, a fim de obter sua posição geográfica corrente, no momento de enviar o resultado de sua transação, executada no ambiente de rede fixa. O banco de dados de localização dos clientes móveis é atualizado por um sistema de controle de localização de clientes móveis responsável por acompanhar a posição geográfica dos clientes móveis, em deslocamento. Essa transação é definida através das operações {Begin-Transaction-TSMC, [Abort-Transaction-TSMC, End-Transaction-TSMC]}. Essas operações delimitam o início e o fim de possíveis operações de leitura sobre os objetos do banco de dado de localização dos clientes móveis.

**Definição 7:** Seja  $TSMC_i$  uma Transação de Suporte à Mobilidade do Cliente,  $E_l$  um esquema local de um banco de dados de localização de clientes móveis, C-Obj nomes de conjuntos de objetos, L-Proc nomes de conjuntos de procedimentos de recuperação (leitura) dos objetos do banco de dados, onde (C-Obj, L-Proc)  $\in E_l$ . Uma Transação de Suporte à Mobilidade do Cliente  $TSMC_i \in TCM_i$  é uma ordem parcial com relação de ordenação  $\angle_i$ , onde:

1.  $TSMC_i = \{\text{Begin-Transaction-}TSMC_i\} \cup \{p_{ik}[\text{obj}] \mid p_{ik} \in \text{L-Proc}, \text{obj} \in \text{C-Obj}\} \cup \{\text{Abort-Transaction-}TSMC_i, \text{Commit-Transaction-}TSMC_i\}$
2.  $\forall p_{ik} \in \text{L-Proc} ((\{\text{Begin-Transaction-}TSMC_i\} \angle_i p_{ik}) \wedge (p_{ik} \angle_i \{\text{Abort-Transaction-}TSMC_i, \text{Commit-Transaction-}TSMC_i\}))$
3.  $\forall p_{ik} \in \text{L-Proc} \{p_{ik} = \text{read}(\text{obj}), \text{obj} \in \text{C-Obj}, \text{onde } \textit{read} \text{ é um método definido sobre os objetos do banco de dados}\}$

O *item 1* da definição 7 determina os delimitadores de início e término da Transação de Suporte à Mobilidade do Cliente (TSMC), bem como o conjunto de procedimentos (métodos) que podem ser executados sobre os objetos do banco de dados.

O *item 2* da definição 7 especifica que, após o evento de término da transação, nenhum outro procedimento pode ser executado, ou seja, o mesmo não é sucedido por qualquer procedimento.

O *item 3* da definição 7 especifica o método (*read*), que é utilizado sobre os objetos do banco de dados em todas as operações da transação TSMC.

Essa transação possui apenas dependência serial em relação à Transação Coordenadora de Banco de Dados (TCBD).

### Dependência Serial para a TSMC ( $TSMC_i \text{ SD } TCBD_i$ )

- $(\text{Begin}_{TSMC_i} \in \text{Histórico-F}) \Rightarrow (\epsilon \rightarrow \text{Begin}_{TSMC_i})$  onde  $\epsilon \in \{\text{Commit}_{TCBD_i}, \text{Abort}_{TCBD_i}\}$

Essa dependência significa que a Transação de Suporte à Mobilidade do Cliente (TSMC) só deve iniciar a sua execução após a conclusão da Transação Coordenadora de Banco de Dados (TCBD).

### Regra de Visibilidade para a TSMC

- $\text{ViewSet}_{TSMC_i} = (\text{BD de localização de } \textit{clientes m\acute{o}veis})$

Essa regra significa que a Transação de Suporte à Mobilidade do Cliente (TSMC) opera diretamente sobre os objetos do banco de dados de localização de clientes móveis, através da operação de leitura (*read*), visualizando o último estado confirmado (*commit*) desse banco de dados.

### 4.3.8

#### A Transação Local (TL)

Uma *Transação Local (TL)* é uma transação componente da *Transação Coordenadora de Banco de Dados (TCBD)*, projetada para executar no banco de dados local, residente nos equipamentos da rede fixa. Essa transação possui um conjunto de operações sobre os objetos do banco de dados local. É definida explicitamente através dos comandos {Begin-Transaction-TL, {Commit-Transaction-TL, Abort-Transaction-TL}}, delimitadores do escopo da transação.

**Definição 8:** Seja  $TCBD_i$  uma Transação Coordenadora de Banco de Dados,  $E_l$  um esquema local de um banco de dados, C-Obj nomes de conjuntos de objetos, C-Proc nomes de conjuntos de procedimentos, onde  $(\text{C-Obj}, \text{C-Proc}) \in E_l$ . Uma Transação Local  $TL_i \in TCBD_i$  é uma ordem parcial com relação de ordenação  $\angle_i$ , onde:

1.  $TL_i = \{\text{Begin-Transaction-}TL_i\} \cup \{p_{ik}[\text{obj}] \mid p_{ik} \in \text{C-Proc}, \text{obj} \in \text{C-Obj}\} \cup \{\text{Abort-Transaction-}TL_i, \text{Commit-Transaction-}TL_i\}$
2. Para qualquer duas solicitações de procedimentos  $(p_{ik}[\text{obj}], p_{jl}[\text{obj}]) \in \text{C-Proc}$ , se **conflitam**  $(\langle \text{obj}, p_{ik} \rangle, \langle \text{obj}, p_{jl} \rangle)$ , então  $(p_{ik}[\text{obj}] \angle_i p_{jl}[\text{obj}]) \vee (p_{jl}[\text{obj}] \angle_i p_{ik}[\text{obj}])$
3.  $\forall p_{ik} \in \text{C-Proc} ((\{\text{Begin-Transaction-}TL_i\} \angle_i p_{ik}) \wedge (p_{ik} \angle_i \{\text{Abort-Transaction-}TL_i, \text{Commit-Transaction-}TL_i\}))$

O *item 1* da definição 8 determina os delimitadores de início e término da Transação Local (TL), bem como o conjunto de procedimentos (métodos) que podem ser executados sobre os objetos do banco de dados.

O *item 2* da definição 8 especifica a ordem em que as operações conflitantes devem ser executadas.

O *item 3* da definição 8 especifica que, após o evento de término da Transação Local (TL), nenhum outro procedimento pode ser executado, ou seja, ele não é sucedido por qualquer procedimento.

A seguir, estão definidas as dependências para execução da Transação Local (TL), nos bancos de dados localizados na rede fixa.

#### Dependência de Cancelamento para a TL ( $TL_i$ AD $TCBD_i$ )

- $(\text{Abort-}TCBD_i \in \text{Histórico-F}) \Rightarrow (\text{Abort-}TL_i \in \text{Histórico-F}) \vee ((\text{Commit-}TL_i \in \text{Histórico-M}) \wedge (\text{Commit-}TL_i \in \text{Histórico-M}))$

Essa dependência, que na verdade é um *axioma* do modelo proposto, significa que, se a *Transação Coordenadora de Banco de Dados*  $TCBD_i$  for cancelada, a transação  $TL_i$  também o será.

#### Dependência Serial para a TL ( $TL_{ij}$ SD $TL_{ij-1}$ )

- $(\text{Begin-}TL_{ij} \in \text{Histórico-F}, 1 < j \leq n) \Rightarrow (\varepsilon \rightarrow \text{Begin-}TL_{ij})$  onde  $\varepsilon \in \{\text{Commit-}TL_{ij-1}, \text{Abort-}TL_{ij-1}\}$

Essa dependência significa que, em uma *Transação Coordenadora de Banco de Dados*  $TCBD_i$ , uma *Transação Local*  $TL_{ij}$  não pode iniciar a sua execução até que a transação  $TL_{ij-1}$  que a precede tenha sido confirmada (*commit*) ou cancelada (*abort*).

As *Transações Locais* não possuem as dependências de confirmação, de término, de exclusão, de confirmação forçada no cancelamento, de início, de início de confirmação e de início de cancelamento.

#### Regra de Visibilidade para a TL

- $\text{ViewSet}_{TL_i} = \{BD_i\}$

Essa regra significa que a Transação Local  $TL_i$  opera diretamente sobre os objetos do banco de dados, através de operações definidas no seu esquema local, visualizando o último estado confirmado (*commit*) do banco de dados.

#### 4.4

#### Critério de Corretude do Modelo Proposto

Conforme enunciado no capítulo 2, *uma transação é uma abstração que representa a seqüência de operações de banco de dados resultante da execução de um programa de aplicação*. É por meio do conceito de transação que se representa a transição de estados de um banco de dados.

Um escalonamento (*schedule*)  $E$  de  $n$  transações  $T_1, T_2, \dots, T_n$  é um ordenamento das operações das transações sujeitas à restrição de que, para cada transação  $T_i$  que participa em  $E$ , as operações de  $T_i$  em  $E$  devem aparecer na mesma ordem em que ocorrem em  $T_i$ . Escalonamentos são ditos *seriais* quando as operações de cada transação são executadas em série, consecutivamente, sem quaisquer operações entrelaçadas de outra transação. Em um escalonamento serial, transações inteiras são executadas em ordem serial:  $T_1$  e em seguida  $T_2$ , e em seguida  $T_3$ , e assim por diante. Assim, pode-se afirmar que um *escalonamento serial é correto*, pois a execução de suas transações sempre leva um banco de dados de um estado consistente para outro estado também consistente [7], [20], [62].

Conforme enunciado nos axiomas 1, 2 e 3, os bancos de dados móveis considerados neste trabalho não possuem interseções entre si e possuem uma estratégia de carga e sincronismo próprio (axiomas 4, 5 e 6) com os bancos de dados localizados na rede fixa, possibilitando a utilização de escalonamentos seriais, conforme explicado na seção 4.2.

O modelo proposto nesta tese trabalha com um escalonamento *serial*, tanto no processamento das transações no ambiente móvel quanto no ambiente de rede fixa. As transações em cada equipamento móvel (cliente móvel) são executadas de forma serial, pois existe uma *Dependência Serial entre as Transações Móveis (TMs)*, conforme apresentado na subseção 4.3.1 deste capítulo. Assim, o processamento local é serial e garantido pelas dependências entre as transações. Uma implementação da garantia da serialização entre as TMs será apresentada no capítulo 5, na seção 5.4.

Vale ressaltar que os escalonamentos são seriais em cada equipamento móvel e que o processamento das transações não apresentará conflitos quando de suas execuções nos bancos de dados da rede fixa, em função dos axiomas 1, 2 e 3, definidos no início deste capítulo. Uma implementação da garantia da serialização entre as *Transações Locais (TLs)*, no equipamento localizado na rede fixa, será também apresentada no capítulo 5, na seção 5.4.

Na prática, o processamento das TMs de cada equipamento móvel nos



bancos de dados localizados na rede fixa criará várias filas de processamento, uma para cada cliente móvel, filas essas que executarão as suas transações de forma serial, na mesma ordem em que foram executadas no equipamento móvel. O processamento de transações de filas diferentes não será conflitante em função dos axiomas 1, 2 e 3 (ausência de sobreposição entre bancos de dados móveis) e elas poderão ser executadas de forma concomitante. O componente de software que irá implementar a *Transação Coordenadora de Banco de Dados* (TCBD) garantirá a seriação de cada escalonamento gerado nos equipamentos móveis quando do seu processamento na rede fixa.

*Logo, o modelo de transações proposto neste trabalho pode ser considerado correto. A seriação do modelo de transações é global e é garantida pelas condições de dependência formalizadas para cada transação do modelo, especificadas anteriormente neste capítulo e que serão implementadas no framework proposto no capítulo 5.*

## 4.5

### Vantagens do Modelo Proposto

Dentre as principais *vantagens* do modelo proposto destacam-se:

1. Não necessitar de conexão contínua do equipamento móvel com a rede fixa, durante o processamento da transação;
2. Garantir o processamento das transações no equipamento da rede fixa, mesmo em momentos de desconexão do equipamento móvel;
3. Minimizar o cancelamento em cascata das transações locais, já confirmadas nos equipamentos móveis, decorrente de sua propagação para o ambiente de rede fixa, em função de que a única hipótese para seu cancelamento será por divergências entre os esquemas dos bancos de dados localizados em cada ambiente (*axioma 10*);
4. Possibilitar a consistência local e global dos dados em menores intervalos de tempo do que os modelos de transações tradicionais, uma vez que o processamento das transações, nos SGBDs localizados na rede fixa, *não necessita de conexão contínua com o equipamento móvel*, mesmo com escalonamentos seriais no processamento de suas transações;
5. Permitir visibilidade dos resultados das transações confirmadas localmente para as transações em execução no equipamento móvel, embora a transação ainda não tenha sido executada na rede fixa;

6. Assegurar a consistência dos dados através da garantia das propriedades ACID para as transações móveis;
7. Possibilitar o processamento de transações curtas e de transações longas, não gerando *overhead* na computação móvel, uma vez que as transações longas podem ser executadas em momentos de desconexão;
8. Possibilitar o uso de SGBDs distribuídos na rede fixa;
9. Acompanhar o *handoff* dos clientes móveis durante o processamento das transações, através das TSMC, visando à entrega dos resultados das transações aos clientes móveis; preservando a segurança e a consistência dos dados.

## 4.6

### Resumo do Capítulo

Este capítulo apresentou o modelo de transações proposto nesta tese, seus axiomas, os estados que suas transações podem assumir durante a execução, sua formalização em lógica de primeira ordem, seu critério de correteude e as vantagens que o modelo proposto apresenta.

No próximo capítulo, será apresentada uma arquitetura para operacionalização e implementação desse modelo de transações no ambiente da computação móvel, usando as tecnologias de framework e de agentes de software.