

## MiD-Mobile: Middleware distribuído para adaptação e gerenciamento de transações em ambiente de computação móvel

Daniela Eloise Flôr<sup>1,2</sup>, Maurício Gonçalves Vieira Ferreira<sup>1</sup>, Nilson Sant'Anna<sup>1</sup>,  
Ricardo Vassela Paulino<sup>2</sup>, Yandre Maldonado e Gomes da Costa<sup>3</sup>

<sup>1</sup>Instituto Nacional de Pesquisas Espaciais (INPE) – São José dos Campos, SP – Brasil

<sup>2</sup>Universidade Paranaense (UNIPAR) – Paranavaí, PR – Brasil

<sup>3</sup>Universidade Estadual de Maringá (UEM) – Maringá, PR – Brasil

danielaflor@unipar.br, mauricio@ccs.inpe.br, nilson@lac.inpe.br,  
ricardovasselaipaulino@gmail.com, yandre@din.uem.br

**Abstract.** *The inherent challenges to the technological development of quality are also applied to the mobile computing systems. In this area, a lot of researches lead to the database transactions. A harmonious accommodation among applications based on transaction processing and dynamic variables of the scenario, motivated the MiD-mobile. The MiD-mobile, monitors such variables and equips with the execution plan of context-aware adapted mobile transaction. In general, the middleware leaves the software to worry with their primary activities, while it uses protocols and services that guarantee context-aware, reduces the interaction with the user, use conscious of resources, among others.*

**Resumo.** *Os desafios inerentes ao desenvolvimento tecnológico de qualidade também se aplicam aos sistemas de computação móvel. Nessa área, muitas pesquisas conduzem às transações de banco de dados. Uma acomodação harmoniosa entre aplicações baseadas em processamento de transação e especificidades dinâmicas do ambiente, incentivaram o MiD-Mobile. O MiD-Mobile, monitora tais especificidades e as equaciona com o plano de execução das transações móveis adaptadas ao contexto. Em âmbito geral, o middleware dá suporte para a atuação primária do software, empregando protocolos e serviços que se preocupam em prover sensibilidade ao contexto, diminuição do ônus do usuário, uso consciente de recursos, entre outros.*

## 1. Introdução

A computação móvel é comumente descrita como uma ampliação da computação distribuída. É caracterizada por uma arquitetura composta de computadores fixos, móveis e meio de comunicação com e sem fio. Os computadores fixos são conectados por cabos e formam a rede fixa, alguns são equipados com interface de comunicação sem fio (*wireless*) e denominados estações de base. As estações de base gerenciam os dispositivos móveis que estão dentro de sua área de abrangência. Os computadores móveis são dispositivos portáteis que se movimentam livremente e por intermédio das estações de base, acessam a rede fixa via conectividade sem fio.

O cenário descrito adiciona aos elementos tradicionais da computação distribuída a comunicação sem fio, a mobilidade e a portabilidade. A portabilidade veio com a miniaturização dos computadores, mas resultou em dispositivos com restrição de recursos. A comunicação sem fio apesar de oferecer liberdade, não garante a mesma qualidade de serviço que as redes cabeadas. Já a mobilidade dos usuários acrescenta novas questões como, por exemplo, a gerência de localização de usuários móveis.

Apesar das limitações é crescente o interesse em sistemas de informação que processem transações nesse contexto, mas a complexidade no desenvolvimento de aplicações fortemente sujeitas a requisitos não funcionais maximiza potencialmente as falhas e dificulta o desenvolvimento de softwares fidedignos por construção.

Outra reflexão deve-se a migração de dados para dispositivos móveis trabalharem autonomamente. A manipulação de dados em equipamentos temporariamente desconectados da rede fixa impacta na forma de gerenciá-los. Vários aspectos de gerência de dados estão sendo revistos, este trabalho enfoca o processamento das transações.

A seção introdutória é seguida por outras cinco seções. A seção seguinte oferece um panorama sobre os desafios do desenvolvimento tecnológico de qualidade em sistemas de computação móvel. Na seção 3 são tratados os modelos de transação móvel e as questões que destinam uma transação ao fracasso. A seção 4 apresenta as transações móveis adaptadas ao contexto, os itens do plano de execução de uma  $t_{mac}$  e os detalhes da arquitetura do MiD-Mobile. A seção 5 engloba os modelos de negócio e o estudo de caso, a última seção exhibe as conclusões e trabalhos futuros.

## 2. Desenvolvimento Tecnológico de Qualidade dos Sistemas de Computação Móvel

Empenhados em solidificar questões de pesquisa importantes para a ciência e para o país, além de agrupar pesquisadores interessados nesses assuntos, a Sociedade Brasileira da Computação, a exemplo de outras, vêm empregando esforços para divulgar os desafios de pesquisa em computação para o decênio 2006/2016.

A computação ubíqua é um dos temas pontuados, ela surge com a convergência da computação móvel e de ambientes inteligentes. Questões em aberto sobre a computação móvel complicam a alavancagem da computação ubíqua. Assim, o presente trabalho, em consonância com os direcionamentos encontrados no relatório do seminário “Grandes Desafios da Pesquisa em Computação no Brasil”, explora alguns desafios para o desenvolvimento tecnológico de qualidade dos sistemas de computação móvel.

Dentre os desafios de pesquisa em computação móvel com vistas à computação ubíqua, encontram-se: (a) a sensibilidade ao contexto, (b) conexão espontânea, transparente e dinâmica, (c) formas diferenciadas de interação com dispositivos, (d) segurança, (e) melhoria na capacidade dos dispositivos, (f) validação, privacidade e valor da informação em contextos inteligentes, (g) disponibilidade, (h) escalabilidade, (i) correteude, (j) persistência, entre outros.

Ao lançar um olhar longitudinal a esses desafios, é possível verificar agrupamentos de interesses como, por exemplo, itens que objetivam atender os fatores humanos, quer seja através de (c) novas formas de interação como fala e gestos, (b) mudança de conexão sem necessidade de intervenção do usuário ou (f) aquisição e manutenção das informações obtidas em contextos inteligentes.

Outra associação reúne os itens preocupados com a arquitetura das aplicações, como (g) disponibilidade, (h) escalabilidade e (a) sensibilidade ao contexto. Os cuidados com a aplicação são prestados pelos itens (d) segurança, (i) correteude, (g) disponibilidade e (j) persistência. Com o intuito de avaliar a correteude das operações e a persistência dos dados dos sistemas de computação móvel, o conceito de transação é visto a seguir.

### 3. Transações

Para [Ferreira e Finger, 2000], uma transação é um modelo de comportamento que assegura a integridade dos dados de um sistema de banco de dados. Tradicionalmente, as transações aplicam operações de leitura e escrita aos dados, operando com atomicidade de execução, isolamento de transações concorrentes, consistência às regras e persistência de resultados, caracterizando as propriedades conhecidas pelo acrônimo ACID.

Os pressupostos teóricos que garantem o sucesso de transações centralizadas ou distribuídas, como, por exemplo, o protocolo de efetivação em duas fases, não se ajustam às freqüentes desconexões do ambiente móvel. Assim, é importante estudar a atitude das transações no cenário da computação móvel e avaliar os modelos de comportamento encontrados na literatura científica.

#### 3.1 Modelos transacionais para ambiente de computação móvel

Genericamente, é desejável que uma transação em ambiente de computação móvel, ou transação móvel, seja capaz de suportar conexões intermitentes, apoiar a mobilidade do cliente durante a transação, atualizar seu deslocamento em função do movimento da estação móvel, manter transações de longa duração, fornecer autonomia em desconexões temporárias, reintegrar atualizações locais na rede fixa, entre outras coisas.

Diante da pluralidade de exigências impostas pelo meio, vários modelos de transação móvel foram propostos, uma vasta revisão do assunto é apresentada em [Chrysanthis, 1993], [Serrano-Alvarado, Roncancio e Adiba, 2004a], [Rocha e Toledo, 2005] e [Côrtes, 2004]. Sem o intuito de esgotar o assunto, os modelos serão agrupados e explorados em suas características determinantes.

Os autores usam critérios diferentes para agrupar os modelos, o que considera a plataforma de execução das transações, o que observa a atuação das propriedades ACID e o que contempla as interferências em sistemas de computação móvel.

[Serrano-Alvarado, Roncancio e Adiba, 2004a] consideram o processamento da transação móvel no equipamento fixo, no dispositivo móvel ou compartilhado entre ambos. Os modelos Kangaroo [Dunham, 1997], Pre-serialization [Dirckze, 2000], Moflex [Ku e Kim, 2000] e MDSTPM [Yeo e Zaslavsky, 1994] indicam o processamento da transação na rede fixa após a requisição ter partido da estação móvel.

Outra possibilidade de processamento, mais autônoma, permite a execução das transações no computador móvel, com a incorporação das alterações na rede fixa. Essa opção agrupa vários modelos como HiCoMo [Lee e Helal, 2002], Pro-motion [Walborn e Chrysanthis, 1998], Clustering [Pitoura e Bhargava, 1995] e [Pitoura e Bhargava, 1999], Two-tier replication [Gray et al., 1996], IOT [Lu e Satyanarayanan, 1994], Prewrite [Madria 2001] e o modelo Semantics-based [Walborn e Chrysanthis, 1995].

A distribuição do processamento da transação entre a rede fixa e o dispositivo móvel operando conectado é considerada pelo modelo Reporting [Chrysanthis, 1993], Clustering [Pitoura e Bhargava, 1995] e Two-tier replication [Gray et al., 1996].

[Rocha e Toledo, 2005] investigam os modelos considerando as desconexões, a largura de banda, as limitações de energia, a escassez de espaço em disco e a mobilidade. Técnicas de caching de dados são empregadas para suportar desconexões, a exceção são os modelos Kangaroo e Pré-serialization. Quando a largura de banda está alta, o modelo IOT sugere sua exploração para transferência de dados, já o Pro-motion e o Semantics-based despacham apenas dados imprescindíveis no canal de comunicação.

Os modelos que consideram o consumo de energia sugerem a divisão da computação, pelo modelo Reporting, o modo de operação desconectado, pelo modelo Semantics-based, e o desligamento da unidade móvel, pelo modelo Kangaroo. A mobilidade é abordada pelos modelos Kangaroo e MDSTPM que geram subtransações enviadas às estações de base por onde os usuários passam.

A escassez de espaço em disco reflete na quantidade de dados enviados às caches das unidades móveis. Os modelos Semantics-based e Pro-motion transferem pequenos fragmentos, já o modelo Reporting sugere a computação distribuída entre a unidade móvel e a rede fixa para diminuir os dados migrados para as caches.

As propriedades ACID são maleabilizadas como forma de viabilizar as transações móveis. [Côrtes, 2004], mostra que a flexibilização da atomicidade vêm da utilização de transações aninhadas ou com a ocorrência de operações de divisão. O resultado é o mesmo, ou seja, subtransações efetivadas mesmo que as outras não sejam, modelos que atendem essa característica são Pro-motion, Reporting, Kangaroo, Moflex, HiCoMo.

Os modelos Pro-motion e Semantics-based agregam formas de acesso consistente aos dados. O Moflex, HiCoMo e o Two-tier replication usam regras que dizem se a transação foi executada corretamente. O isolamento é flexibilizado pela liberação total ou gradual de resultados parciais, proposto nos modelos IOT, Reporting, HiCoMo, Pré-serialization, Semantics-based e Pro-Motion. A durabilidade é prejudicada pela inconstância do meio, o modelo HiCoMo permite a persistência de subtransações.

É perceptível a inexistência de um modelo de transação único que satisfaça todas as variáveis que influenciam em um sistema de computação móvel. Em vista disso, é interessante que diferentes estratégias de atuação de uma transação estejam aptas a operar, aumentando as chances de a transação obter êxito e do software obter qualidade.

A atuação diversificada das transações é possível através da adaptação. Adaptar, segundo [Coulouris, Dollimore, Kindberg, 2007], significa modelar soluções computacionais em função de certos fatores. Sendo assim, é necessário identificar esses fatores e verificar a influência que exercem sobre as transações.

### 3.2 Adaptação de transações em função do ambiente de computação móvel

Sistemas de informação que processam transações e operam em ambiente de computação móvel devem coexistir com a conectividade sem fio, com a portabilidade dos dispositivos e com a mobilidade dos usuários e ainda sim concluírem suas ações satisfatoriamente. O mesmo tripé que sustenta a computação móvel restringe as transações de banco de dados nesse ambiente.

Os dispositivos móveis são personificados em assistentes pessoais digitais (PDA), telefones celulares, telefones inteligentes (*smart phones*), entre outros. Normalmente possuem recursos limitados no que diz respeito a espaço em disco, poder de processamento e suprimento autônomo de energia. Avanços tecnológicos são constantes e vão continuar popularizando equipamentos com recursos e funções heterogêneas.

A heterogeneidade e as limitações dos dispositivos pedem uma avaliação no tamanho dos dados disponíveis localmente, na capacidade computacional das estações móveis, além da adoção de diferentes modos de operação para economia de energia.

Modos mais maleáveis de operação propostos em [Ahmed et al., 1999], também atendem a variabilidade da comunicação sem fio. Quando a rede estiver descongestionada é o período ideal para uma estação móvel estar fortemente conectada à rede fixa, transferindo dados. Uma conexão parcial pode ser a saída quando a largura de banda for suficiente para receber dados. Quando o canal de comunicação estiver indisponível trabalhar desconectado é uma opção para que as estações móveis continuem a operar.

O modo soneca é indicado quando a estação móvel não tem ou está no limite dos seus recursos, um estado de vigília é mantido para receber o retorno das requisições enviadas a rede fixa.

Uma consideração importante sobre o enlace de rede é a variação da intensidade do sinal. Momentaneamente, dispositivos portáteis em movimento tentam acessar a rede estando em áreas de sombra. Essas áreas são espaços geográficos com cobertura insuficiente ou completamente sem abrangência da comunicação sem fio, impedindo o cliente de usar recursos disponíveis na rede fixa, podendo trabalhar apenas localmente.

Outra questão é a diversidade de tecnologias de comunicação sem fio usadas na computação móvel, como Bluetooth, Wi-Fi e GPRS, cada uma com suas especificidades.

Todas as considerações dessa seção devem ser ponderadas pelos desenvolvedores de solução computacional, somam-se a essas as preocupações tradicionais com os requisitos da aplicação, prazos, custos e outras. Aumentar a lista das dificuldades comuns à criação de software contribui para o prejuízo do seu desenvolvimento.

A singularidade de propósito identificada nos modelos de transação estudados comparada à pluralidade de interferências que afetam os sistemas de computação móvel requereu a aplicação de estratégias mais dinâmicas como a adaptação, a fim de reutilizar o software em contextos múltiplos.

Diante do exposto se faz oportuna a proposta de um middleware com funções de monitoramento dessas interferências e de adaptação, em tempo de execução, frente às mudanças monitoradas. Sistemas adaptativos possuem comportamento vinculado ao ambiente em que estão inseridos, essa estratégia é explorada em várias áreas da Ciência da Computação, inclusive no processamento de transação para a computação móvel.

[Serrano-Alvarado, 2004b] propôs um Serviço de Transações Móveis (STM) que considera algumas informações a respeito do dispositivo móvel e da rede fixa para sugerir alternativas de execução para as transações móveis. As alternativas de execução são aplicadas de acordo com descritores pré-fixados do ambiente.

O projeto de [Serrano-Alvarado, 2004b] pressupõe uma configuração capaz de suportar um SGBD nos dispositivos móveis, dificultando o acesso à aplicação por uma gama de clientes com equipamentos de configuração modesta. A proposta inclui serviços às estações de base ao fazer uso da arquitetura cliente/agente/servidor, essa decisão implica em um gargalo que atrapalha a escalabilidade da aplicação. Outra consideração é sobre o modo de operação, o STM considera apenas o modo conectado e desconectado.

[Rocha e Toledo, 2007] empregam uma adaptação colaborativa entre as transações e a Plataforma de Gerenciamento de Transações Adaptáveis (PGTA). A plataforma é responsável por monitorar recursos de conectividade e do dispositivo, além de flexibilizar a propriedade do isolamento. Às transações cabe manter as políticas de adaptação necessárias para reagir às mudanças do ambiente.

A proposta prevê três modos distintos de operação, (R) remoto, (L) local e (LR) local-remoto. As opções L e LR usam dados em cache, sendo que o modo L não mantém acesso exclusivo aos dados remotos, implicando na inconsistência dos mesmos. A PGTA considera que todos os objetos necessários para aplicar o modo L ou LR estão disponíveis em cache, o que dependendo do domínio da aplicação pode ser inviável.

O middleware MiD-Mobile idealizado nessa pesquisa visa atender diferentes domínios de aplicação através da harmonização entre a necessidade e a disponibilidade de recursos computacionais, sem sacrificar as funcionalidades de cada software.

O MiD-Mobile promove a adaptação considerando três dimensões gerais, a conectividade sem fio, os recursos e os “dados” disponíveis no dispositivo móvel. A figura 1, inspirada em [Özsu e Valduriez, 2001], enfatiza três configurações passíveis de adaptação para uma transação.

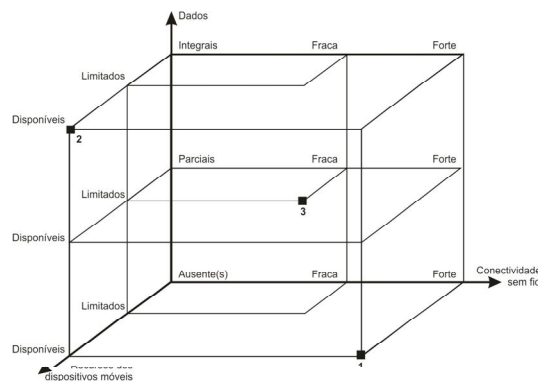


Figura 1. Dimensões consideradas para a adaptação das transações



O ponto numerado como 1 representa uma situação em que o dispositivo móvel apesar de apto a exercer processamento e de usufruir de boa conectividade, não executa a transação por não ter dados locais, passando a execução para a rede fixa. O segundo ponto exemplifica o processamento local da transação, quando há recursos e dados disponíveis na estação móvel, mas a conectividade deixa a desejar.

O ponto 3 ilustra o processamento da transação distribuído entre dispositivo móvel e rede fixa, esse exemplo ocorre quando os dados em cache são incompletos, ou quando a conectividade é insuficiente para efetuar uma transferência de dados ou quando os recursos estão no limite, suportando apenas computação limitada.

As contribuições do MiD-Mobile em relação aos trabalhos pesquisados dizem respeito à inclusão da dimensão “dados” para efetuar a adaptação, a isenção de configuração mínima dos dispositivos no acesso às aplicações e a admissão da influência de estados diferentes da conectividade sem fio ocasionada, por exemplo, pela intensidade do sinal. Os demais trabalhos analisados são de teor acadêmico, dificultando comparações práticas sobre os resultados, permitindo apenas questões conceituais.

Para oferecer suporte a essas contribuições são necessários algoritmos, protocolos e serviços específicos que ajustem as transações ao plano de execução (PE) mais promissor. Quando o MiD-Mobile recebe as transações requisitadas pela aplicação, várias atividades são iniciadas para elaborar o PE, somente após essa definição a transação é submetida aos dados. A próxima seção detalha as transações, os planos de execução e a arquitetura do MiD-Mobile.

#### 4. O MiD-Mobile e as Transações Móveis Adaptadas ao Contexto

Uma transação tratada pelo MiD-Mobile é designada transação móvel adaptada ao contexto ( $t_{mac}$ ). As dimensões gerais consideradas durante a adaptação, ilustradas na figura 1, foram mineradas e minúcias foram identificadas. Na tabela 1, constam as dimensões gerais, os detalhes e os conteúdos discretizados tratados pelo middleware.

**Tabela 1. Aspectos avaliados durante a adaptação**

Dimensões	Detalhes	Conteúdos
Rede	Estado da conexão	Conectado (0) Desconectado (1)
	Taxa de transferência	Suficiente (0) Intermediária (1) Insuficiente (2)
	Intensidade do sinal	Suficiente (0) Intermediária (1) Insuficiente (2)
Dados	Situação dos dados	Possui dados (0) Não possui dados (1)
	Completo dos dados	Completo (0) Incompleto (1) Indefinido (2)
	Validade dos dados	Dados Válidos (0) Inválidos (1) Indefinido (2)
Dispositivo	Capacidade de Bateria	Suficiente (0) Intermediária (1) Insuficiente (2)
	Capacidade de hardware	Suficiente (0) Intermediária (1) Insuficiente (2)

Uma  $t_{mac}$  agrega à transação o plano de execução gerado em tempo de execução. O Plano de Execução da  $T_{mac}$  é formado pela indicação da plataforma de processamento da transação, pelo modo de operação que o dispositivo móvel pode manter durante a execução da transação e pela aplicação tradicional ou estendida das propriedades ACID. Os itens foram sintetizados na tabela 2.

**Tabela 2. Plano de Execução das  $T_{mac}$ 's**

Dimensões do PE	Conteúdos
Plataforma de execução	Local
	Remoto
	Distribuído
Modo de operação	Totalmente conectado
	Parcialmente conectado
	Desconectado
	Soneca
Propriedades ACID	Atomicidade tradicional ou estendida
	Consistência tradicional ou estendida
	Isolamento tradicional ou estendido
	Durabilidade tradicional ou estendida

A plataforma de execução de uma  $t_{mac}$  pode ser local, quando a transação atua sobre dados antecipados, pode ser remota, quando a transação é enviada para a rede fixa ou pode ser distribuída quando a transação é dividida em subtransações executadas local e remotamente. O controle do modo de operação do dispositivo prevê a economia de recursos sem prejuízo para as aplicações. As condições da rede, dos dispositivos e a ocorrência de desconexões implicam diretamente nas propriedades ACID.

O MiD-Mobile permite configurar o período de desconexão tolerado pela aplicação e monitorar esse tempo enquanto aplica algoritmos elaborados para estender as propriedades ACID, de acordo com a plataforma de execução da  $t_{mac}$ . Na opção remota a durabilidade exige atenção, na opção local a preocupação é com a consistência. Já a divisão das transações requer atomicidade local, global e cautela com o isolamento.

Os algoritmos da figura 2 estendem a atomicidade, consistência e durabilidade e são aplicados pelo middleware, o isolamento é estendido a partir de configuração direta no banco de dados. Os algoritmos foram escritos em português a fim de simplificar o entendimento, alguns comandos foram adicionados e descritos na legenda.

Algoritmo de extensão da Atomicidade (a)	Algoritmo de extensão da Atomicidade (b)	Algoritmo de extensão da Durabilidade (c)
2PCA <sub>i</sub> no Dispositivo Móvel	2PCA <sub>i</sub> na Rede Fixa	Rede Fixa
01 Particiona $t_{mac}$	01 Leia <sub>(STDM)</sub> $t_{mac1}$ e $t_{mac2}$	01 Leia <sub>(STDM)</sub> UM <sub>ID</sub> $t_{mac}$
02 Escreva <sub>(STDM)</sub> $t_{mac1}$ e $t_{mac2}$	02 Monitora SGT <sub>des</sub>	02 Monitora SGT <sub>des</sub>
03 Monitora SGT <sub>des</sub>	03 Executa $t_{mac2}$	03 Executa $t_{mac}$
04 Executa $t_{mac1}$	04 Repita	04 Repita
05 Escreva <sub>(STDM)</sub> $t_{mac}$ (dados)	05 Monitora SGT <sub>des</sub>	05 Monitora SGT <sub>des</sub>
06 Repita	06 Monitora SGC	06 Monitora UM
07 Monitora SGT <sub>des</sub>	07 Se (SGT <sub>des</sub> > $\Delta_t$ ) então	07 Se ( $t_{des} > \Delta_t$ ) então
08 Monitora SGC	08 Se (SGC) então	08 Se (not(UM)) então
09 Se (SGT <sub>des</sub> > $\Delta_t$ ) e not(RF( $t_{mac2}$ ))	09 Executa $t_{mac2/comp2}$	09 Executa $t_{mac/comp}$
10 Executa $t_{mac1/comp1}$	10 Escreva <sub>(STDM)</sub> Executa $t_{mac1/comp1}$	10 Escreve <sub>(STDM)</sub> Mensagem( $t_{mac}$ )
11 Até que ( $t_{des} > \Delta_t$ )	11 Escreva <sub>(STDM)</sub> $t_{mac1}$ (dados)	11 Se (UM) então
	12 Se not(SGC) então	12 Executa $t_{mac/comp}$
<b>Algoritmo de extensão da concorrência (d)</b>	13 Executa $t_{mac2/comp2}$	13 Escreve <sub>(STDM)</sub> Mensagem ( $t_{mac}$ )
Controle de Concorrência Pessimista na Rede Fixa	14 Escreva <sub>(STDM)</sub> $t_{mac1/comp1}$	14 Se ( $t_{des} \leq \Delta_t$ ) e (not(UM)) então
01 Leia <sub>(STDM)</sub> (UM <sub>ID</sub> $t_{mac}$ , $t_{mac}$ (dados), $t_{mac}$ (dados), $t_{des}$ )	15 Escreva <sub>(STDM)</sub> $t_{mac1}$ (dados)	15 Monitora SGT <sub>des</sub>
02 Se (SGT <sub>des</sub> > $\Delta_t$ ) e (UM( $t_{mac}$ (dados))) = RF( $t_{mac}$ (dados)))	16 Se (SGT <sub>des</sub> <= $\Delta_t$ ) e (SGC) então	16 Se (SGT <sub>des</sub> <= $\Delta_t$ ) e (UM) então
03 Leia $t_{mac}$ (dados)	17 Executa $t_{mac1}$	17 Escreve <sub>(STDM)</sub> Mensagem ( $t_{mac}$ )
04 Executa $t_{mac}$	18 Se ( $t_{mac1}$ ) então	18 Se ( $t_{mac}$ ) então
05 Escreva $t_{mac}$ (dados)	19 Executa <sub>(RF/UM)</sub> 2PCA <sub>i</sub>	19 Commit ( $t_{mac}$ )
06 Escreva <sub>(STDM)</sub> commit( $t_{mac}$ )	20 Escreva $t_{mac1}$ (dados)	20 Escreve <sub>(STDM)</sub> Mensagem ( $t_{mac}$ )
07 Senão	21 Se (not( $t_{mac1}$ )) então	21 $T_{des} := X > \Delta_t$
08 Escreva <sub>(STDM)</sub> $t_{mac/comp}$	22 Executa $t_{mac2/comp2}$	22 Se (not( $t_{mac}$ )) então
09 Escreva $t_{mac}$ (dados)	23 Escreva <sub>(STDM)</sub> Executa $t_{mac1/comp1}$	23 Executa $t_{mac/comp}$
	24 Escreva <sub>(STDM)</sub> $t_{mac1}$ (dados)	24 Escreve <sub>(STDM)</sub> Mensagem ( $t_{mac}$ )
	25 $t_{des} := X > \Delta_t$	25 $t_{des} := X > \Delta_t$
	26 Até que ( $t_{des} > \Delta_t$ )	26 Até que ( $t_{des} > \Delta_t$ )



<p><b>Algoritmo de extensão da concorrência (e)</b> Controle de Concorrência Otimista na Rede Fixa</p> <p>01 Leia<sub>(STDM)</sub> (UM<sub>ID</sub>, t<sub>mac</sub>, t<sub>A</sub>(t<sub>mac</sub>dados), t<sub>r</sub>, N(t<sub>mac</sub>dados), t<sub>des</sub>)</p> <p>02 Leia t<sub>mac</sub>dados</p> <p>03 Executa t<sub>mac</sub></p> <p>04 Se (SGT<sub>des</sub> &lt; Δ<sub>t</sub>) e (UM(t<sub>r</sub>, A(t<sub>mac</sub>dados)) = RF(t<sub>r</sub>, A(t<sub>mac</sub>dados)))</p> <p>05 Escreva t<sub>r</sub>, N(t<sub>mac</sub>dados)</p> <p>06 Escreva<sub>(STDM)</sub> commit(t<sub>mac</sub>)</p> <p>07 Senão</p> <p>08 Executa t<sub>mac/comp</sub></p> <p>09 Escreva<sub>(STDM)</sub> RF(t<sub>r</sub>, N(t<sub>mac</sub>dados))</p>	<p><b>LEGENDA</b></p> <p>Unidade móvel (UM)</p> <p>Identificador da UM (UM<sub>ID</sub>)</p> <p>Rede fixa (RF)</p> <p>Tempo predefinido (Δ<sub>t</sub>)</p> <p>Tempo de desconexão (t<sub>des</sub>)</p> <p>Transação móvel adaptada ao contexto (t<sub>mac</sub>)</p> <p>Transações compensatórias de t<sub>mac</sub> (t<sub>mac/comp</sub>)</p> <p>Timestamp atual dos dados de t<sub>mac</sub> t<sub>A</sub>(t<sub>mac</sub>dados)</p> <p>Novo timestamp dos dados de t<sub>mac</sub> t<sub>S</sub>N(t<sub>mac</sub>dados)</p> <p>Serviço de troca de dados e mensagens (STDM)</p> <p>Serviço gerenciador do tempo de desconexão (SGT<sub>des</sub>)</p> <p>Particiona: cria subtransações</p> <p>Monitora: acompanha o recurso</p>
--	--

**Figura 2. Algoritmos de extensão das propriedades ACID**

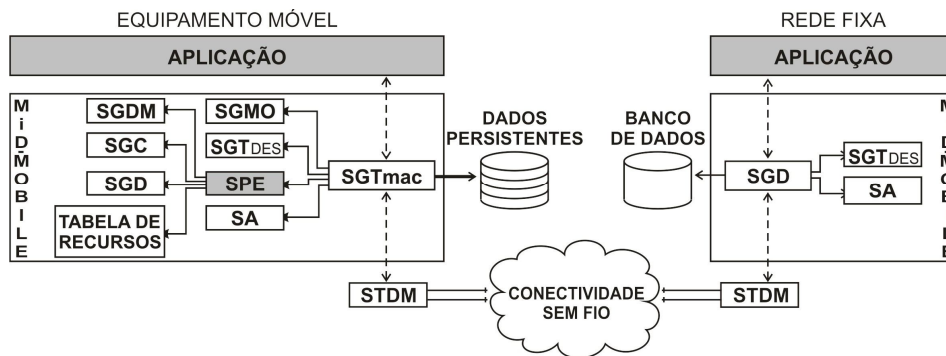
A notação “Escreva<sub>(STDM)</sub> t<sub>S</sub>A(t<sub>mac</sub>dados)” identifica a ação e o local onde a instrução irá ocorrer, o exemplo é um comando de escrita do timbre de hora atual dos dados de t<sub>mac</sub> no serviço de troca de dados e mensagens. Todos os itens inseridos no STDM são encaminhados quando o dispositivo móvel se reconecta à rede fixa.

É possível verificar que a extensão das propriedades acomoda o monitoramento do tempo de desconexão. Na plataforma de execução distribuída, dois algoritmos são propostos para a atomicidade, o que executa na rede fixa e o que executa no dispositivo móvel. Se o cliente ficar desconectado por um tempo maior do que o aceitável o algoritmo (a) linha 9 e o algoritmo (b) linha 26 cancelam a transação.

Na plataforma local, algoritmos de controle de concorrência pessimista ou otimista são usados na sincronização, a opção (d) observa a inconsistência logo na linha 2 e a opção (e) na linha 4. Na execução remota, o algoritmo da durabilidade (c) requer a confirmação dos usuários que as iniciaram, linha 16. Algumas instruções dos algoritmos apresentados usam serviços do middleware, a seção seguinte detalha sua arquitetura.

**4.1 A arquitetura do MiD-Mobile**

O principal objetivo do MiD-Mobile é garantir que as transações requeridas pela aplicação obtenham êxito, para isso, vários serviços foram desenvolvidos, a figura 3 oferece uma visão geral da arquitetura do middleware.



**Figura 3. Arquitetura do MiD-Mobile**

Em sua função principal, o middleware aciona o serviço de gerenciamento de recursos do dispositivo móvel (SGDM), o serviço de gerenciamento da comunicação (SGC) e o serviço de gerenciamento dos dados (SGD). Cada serviço atua junto a uma das dimensões citadas na tabela 1, capturando informações para compor a tabela de

recursos. Dessa tabela constam as entradas necessárias para que o serviço de plano de execução (SPE) decida qual é a estratégia transitória mais adequada à  $t_{mac}$ .

Para que o plano de execução da  $t_{mac}$  possa realmente oferecer chances de êxito, ele é proposto logo que a transação é requerida pela aplicação. O serviço gerenciador de transações móveis adaptadas ao contexto ( $SGT_{mac}$ ) recebe a requisição, invita o serviço gerenciador de tempo de desconexão ( $SGT_{des}$ ) e o SPE. O  $SGT_{des}$  é inicializado para o caso de ser necessária a aplicação estendida de alguma das propriedades ACID. Já o SPE é convocado a organizar o plano de execução.

Após a operação do SPE, cabe ao  $SGT_{mac}$  indicar ao serviço de gerenciamento de modo de operação (SGMO) como o dispositivo móvel pode operar, ao mesmo tempo que economiza recursos e não oferece prejuízos para a aplicação. Outras responsabilidades do  $SGT_{mac}$  incluem gerenciar o processamento local das transações, receber e encaminhar para a aplicação o resultado de execuções remotas e colaborar com o serviço de gerenciamento de transação (SGT) da rede fixa, na tarefa de gerir subtransações locais e remotas.

A comunicação entre o  $SGT_{mac}$  e o SGT é auxiliada pelo serviço de troca de dados e mensagens (STDM). Inclusive, é responsabilidade do STDM encaminhar os dados e as mensagens geradas durante a desconexão, no momento em que o dispositivo móvel se reconecta à rede fixa.

Assim que um plano de execução de uma  $t_{mac}$  for aplicado, o  $SGT_{mac}$  convoca o serviço de aprendizagem (SA) para registrar a ocorrência e os resultados. Essa prática é adotada considerando o fato do SPE ser um mecanismo de classificação baseado em árvore de decisão (AD).

Classificar significa qualificar um exemplo em uma classe. Existem vários métodos de classificação, a escolha por AD deve-se às características como representatividade discreta dos valores dos atributos, inclusive o de classificação, robustez ao verificar ruídos (classificações incorretas) no conjunto de treinamento ou ausência de valores de atributos nas instâncias, entre outros.

A criação de uma árvore de decisão consiste no particionamento recursivo de um conjunto de treinamento, até que cada subconjunto obtido contenha casos de uma única classe. Neste estudo, o conceito de classe diz respeito ao plano de execução da  $t_{mac}$ . A figura 4 ilustra um modelo de AD e seu respectivo conjunto de treinamento.

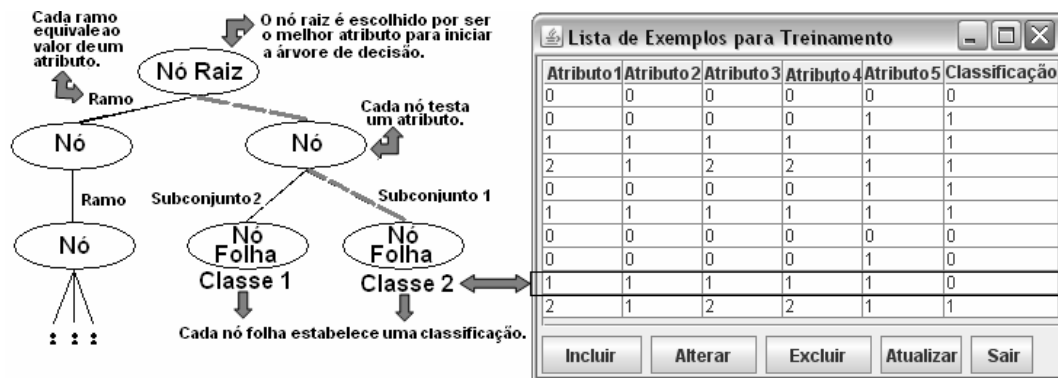


Figura 4. Modelo de árvore de decisão e de conjunto de treinamento

O algoritmo de indução utilizado no MiD-Mobile foi o ID3. A escolha justifica-se pela completude no espaço de busca da árvore, pela seleção estatística (entropia) das instâncias do problema, pelo ganho de desempenho devido à ausência de *backtracking*, ou seja, ausência de reconsideração de atributos anteriores, entre outros.

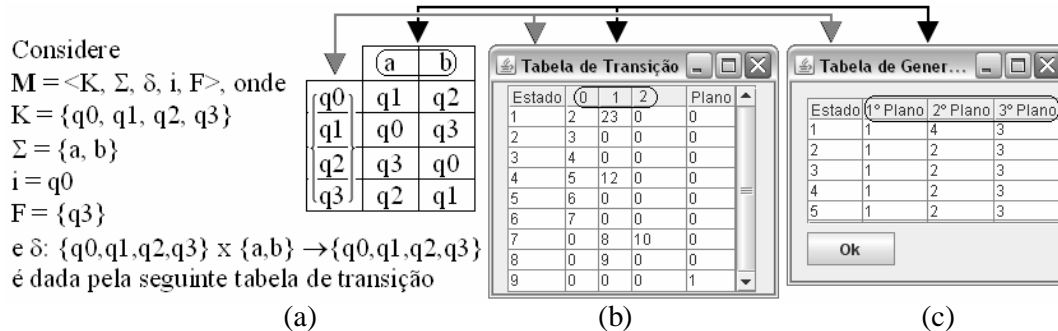
Segundo [Russel e Norvig, 2003], o ID3 é um dos algoritmos pioneiros na indução de árvores de decisão, sua essência consiste na seleção do atributo mais hábil para classificar o exemplo, essa escolha repete-se a cada nó descendente. Uma medida quantitativa, chamada ganho de informação, é usada para verificar quão bem um atributo classifica as instâncias de treinamento.

Todavia, o conjunto de treinamento pode conter classificações incorretas (ruído), o ID3 emprega o cálculo da entropia para medir a impureza do conjunto. A fórmula do ganho de informação verifica o ganho individual de cada atributo em relação às instâncias do conjunto de treinamento, mesmo considerando as impurezas. Uma árvore de decisão é útil se o seu aprendizado permite classificar corretamente exemplos não-treinados, essa é a principal função do SPE.

O ID3 emprega recursividade na escolha do melhor atributo, essa tática faz crescer progressivamente o uso da memória e do processamento, sendo assim, aparentemente, o algoritmo não é uma proposta coerente face à restrição de recursos dos dispositivos. A solução surgiu à luz da teoria dos grafos, que empresta seus elementos gráficos à visualização de árvores de decisão e autômatos finitos determinísticos.

Um autômato finito determinístico é uma quintupla  $\langle Q, \Sigma, \delta, q_0, F \rangle$ , onde  $Q$  é um conjunto finito de estados internos,  $\Sigma$  o alfabeto de entrada,  $\delta$  função de transição,  $q_0$  o estado inicial e  $F$  o conjunto de estados finais. Resumidamente, um AFD lê uma string de entrada que contém símbolos do alfabeto. A função de transição promove a mudança de estado de acordo com o símbolo lido, ao término da leitura diz-se que o AFD aceita a cadeia caso ela tenha partido do estado inicial e atingido um estado final.

Um AFD pode facilmente ser representado por uma tabela de transição, essa característica foi estendida às árvores de decisão. A figura 6 apresenta uma tabela de transição para um AFD  $M$ , uma tabela de transição da árvore de decisão gerada pelo MiD-Mobile e uma tabela de transição usada na generalização de novos casos.



**Figura 5. Tabela de transição para o AFD  $M$  (a), tabela de transição (b) e tabela de generalização (c) gerada pelo MiD-Mobile**

Na tabela de transição (b) os nós da árvore equivalem aos estados do AFD e são descritos na coluna “estado”. Os ramos 0, 1 e 2 tomaram o lugar dos símbolos do

alfabeto. O acréscimo ficou por conta da coluna plano que só é preenchida quando precisar representar um nó folha, ou seja, uma classificação. Classificar casos de teste, ou novos casos, é uma forma de verificar a capacidade de generalização da árvore, ou seja, se ela adquiriu conhecimento ou apenas memorizou a amostra.

Quando uma instância de teste for aplicada à tabela de transição (b), o algoritmo tradicional do AFD pode se perder ao encontrar em seu percurso um estado que não reconhece o símbolo lido. Duas operações complementares foram usadas nessa situação, uma adaptação no algoritmo e a criação de uma tabela de generalização.

O algoritmo do AFD foi adaptado para percorrer a tabela de transição da árvore de decisão até o subconjunto conhecido, em seguida passa para a tabela de generalização e quantifica a incidência de cada plano nos subconjuntos descendentes. Assim, a tabela de generalização difere da tabela de transição por armazenar quantas vezes um determinado nó esteve presente na classificação de um plano e não o próximo atributo a ser visitado.

Percorrer a tabela de transição e de generalização através de um algoritmo de mudança de estado é mais simples e mais econômico em termos de memória e processamento do que o algoritmo de indução da AD. A criação da árvore e dos mecanismos de aprendizagem e generalização é responsabilidade do serviço de aprendizagem (SA) disponível na rede fixa. Ao iniciar o acesso à aplicação, esses conteúdos são enviados para o dispositivo móvel. Assim o SPE pode usufruir de leveza e simplicidade ao pesquisar o plano de execução mais promissor para uma  $t_{mac}$ .

No período mais apropriado para cada domínio de aplicação, as instâncias de testes e os resultados armazenados pelo SA do dispositivo móvel são enviados ao SA da rede fixa, visando novo treinamento. O aproveitamento das experiências colhidas em cada dispositivo permite à árvore melhorar seu desempenho em novos casos. No momento seguinte, os elementos são reenviados ao dispositivo móvel. A próxima seção apresenta uma simulação de um modelo de negócio usando o MiD-Mobile.

## 5. Modelos de Negócio e Estudo de Caso

Todos os esforços destinados a pesquisas em computação móvel como infraestrutura, formas de interação, ferramentas, entre outras questões, são justificados pela variedade de modelos de negócio que podem usufruir de tais avanços.

O foco dessa pesquisa diz respeito aos sistemas de computação móvel que acessam dados compartilhados, nesse contexto encontram-se aplicações, como por exemplo, para agendamento, vendas, serviços ou movimentações bancárias. Uma das áreas que mais está usufruindo dos avanços da computação móvel é a medicina. Nesse campo estão sendo desenvolvidos vários sistemas de computação móvel para hospitais, centros de saúde, atendimento domiciliar e outros.

Para legitimar o middleware proposto, foi desenvolvida uma aplicação que consulta o MiD-Mobile, antes de efetuar uma transferência de fundos, para saber a melhor maneira de executar a tarefa. A figura 6 trás vários momentos da utilização da aplicação, os quadros (a), (b) e (c) mostram um caso de sucesso (commit) e um caso de fracasso (abort) de uma transação, o item (d) apresenta uma entrada aleatória na tabela de recursos do SPE, já o item (e) exhibe o plano de execução da  $t_{mac}$  iniciada em (a).

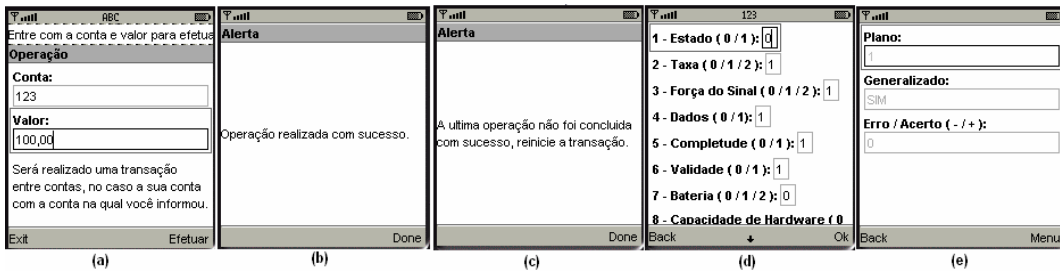


Figura 6. Aplicação de transferência de fundos

No exemplo apresentado na figura 6 - caso (e), o SPE sugeriu o plano de execução remoto da  $t_{mac}$  (plano 1), informou que a instância de teste foi induzida e simulou o insucesso (erro) e o sucesso (acerto) do plano para aquela transação.

Também foi desenvolvido um simulador para o aprendizado na rede fixa, a figura 7, apresenta um ambiente onde os planos de execução obtidos a partir de novos testes podem ser aceitos ou rejeitados. Simular o aceite do plano representa o término satisfatório da transação (commit) e rejeitar significa o seu insucesso (rollback).

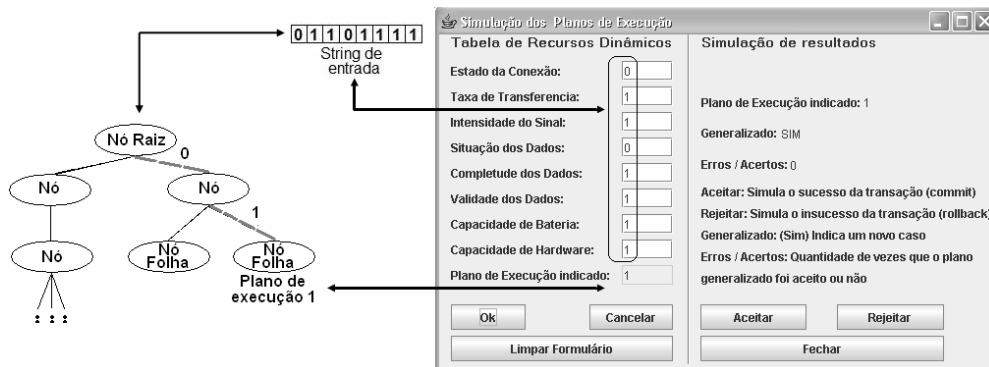


Figura 7. Simulação da generalização de novos casos

O MiD-Mobile ainda permite configurar um limiar de tentativas, ou seja, toda vez que a execução de um plano for rejeitada, um erro é computado e ao atingir o limiar uma mudança de plano é sugerida. Para validar a arquitetura completamente é necessária uma avaliação mais profunda sobre a discretização de valores da tabela de recursos, bem como alguns aspectos na aprendizagem como a manipulação de atributos com diferentes custos (devido a pluralidade de formas de comunicação sem fio) e o tratamento de valores desconhecidos (para considerar dados com completude e validade indefinidos).

## 6. Conclusões e Trabalhos Futuros

Muitos esforços na área da computação móvel, servem de mola propulsora para a concretização da computação ubíqua. Mas algumas questões estão retardando essa efetividade. Este artigo dispensa atenção aos desafios implícitos na execução de transações móveis em ambiente compartilhado, mas muitos outros cenários desafiantes se configuram, principalmente os que adicionam inteligência ao ambiente.

Dentre as questões que podem interferir na construção de sistemas de computação móvel fidedignos por construção, estão a corretude das operações de

utilização e a persistência adequada dos dados. Uma das formas de manter a fidedignidade do software é garantir que ele faça as tarefas a que ele se propõem e que as faça corretamente. Assim, é válido garantir a melhor forma de execução de transações que operam mudanças no contexto da aplicação e a consequente persistência dos dados gerados por tais execuções.

O MiD-Mobile encerra em si essas funções, mas não esgota a pluralidade de temas desafiante como, por exemplo, escalabilidade do software. O middleware objetiva a popularização do acesso à aplicação por dispositivos móveis com modesta configuração, aumentando assim as chamadas à rede fixa. Em pesquisas futuras, faz-se necessário verificar como as aplicações reagirão a isso e como garantir que a disponibilidade de serviço não seja prejudicada.

Novas melhorias nos dispositivos podem requerer novos serviços na arquitetura do middleware, tendo em vista a acomodação da heterogeneidade resultante de dispositivos com funções e recursos mais avançados.

Outra consideração é sobre a desejável onipresença da computação. Quanto mais o MiD-Mobile dispensar a colaboração do usuário, menor será o seu ônus com a aplicação, e mais a evolução dessa tendência coincidirá com a idéia de interação natural e imperceptível do usuário com o ambiente. Para alcançar esse estado a sensibilidade ao contexto é a estratégia que merece atenção. É importante que as aplicações saibam agir dependendo do contexto e da semântica da aplicação.

Muitas outras questões permanecem esperando respostas, sejam elas aplicáveis ao MiD-Mobile ou não. Assim, é válido conhecer e advogar na direção dos desafios previstos para o decênio 2006/2016, além de conhecer outras iniciativas como as que constam de [Hoare e Milner, 2004].

## Referências

- Ahmed, K., Ismail, M., El-Makky, N. e Nagi, K. (1999) "A New Transaction Management Scheme for Mobile Computing Environments". Proceedings of the IASTED, MIT, Boston-USA, November 3-6.
- Chrysanthis, P. K. (1993) "Transaction Processing in Mobile Computing Environment". In IEEE Workshop on APADS, Princeton, USA, October.
- Côrtes, S. C. (2004) "Um modelo de transações para integração de SGBD a um ambiente de computação móvel", 215 p. Tese (Doutorado) – PUC Rio de Janeiro.
- Coulouris, G., Dollimore, J. e Kindberg, T. (2007) "Sistemas distribuídos: conceitos e projeto", tradução João Tortello. 4. ed. Porto Alegre: Bookman, 784 p.
- Dirckze, R. e Gruenwald, L. (2000) "A pre-serialization transaction management technique for mobile multidatabases". MONET, vol. 5, no. 4.
- Dunham, M. H., Helal, A. e Balakrishnan, S. (1997) "A mobile transaction model that captures both the data and movement behavior". ACM/Baltzer Journal on Special Topics in Mobile Networks and Applications, vol.2.
- Ferreira, J. E. e Finger, M. (2000) "Controle de concorrência e distribuição de dados: a teoria clássica, suas limitações e extensões modernas". São Paulo, IME-USP, 179 p.



- Gray, J., Helland, P., O'neil, P. e Shasha, D. (1996) "The dangers of replication and a solution". In ACOMA SIGMOD Conference Montreal, Canada: June.
- Hoare, T. e Milner, R. (2004) "Grand Challenges in Computing", <http://www.bcs.org/upload/pdf/gcresearch.pdf>, maio de 2008.
- Ku, K., Kim, Y. (2000) "Moflex transaction model for mobile heterogeneous multidatabase systems". In IEEE Workshop on Research Issues in Data Engineering, San Diego, USA, Feb.
- Lee, M., e Helal, S. (2002) "HiCoMo High Commit Mobile Transactions". Kluwer Academic Publishers Distributed and Parallel Databases (DAPD), vol . 11, no. 3.
- Lu, Q. e Satyanarayanan, M. (1994) "Isolation-only transactions for mobile computing". In ACM Operating Systems Review, vol. 28, n. 3.
- Madria, S. K. e Bhargava, B. (2001) "A transaction model for improving data availability in mobile computing". Kluwer Academic Publishers Distributed and Parallel Databases (DAPD), vol . 10, no. 2.
- Özsu, M. T. e Valduriez, P. (2001) "Princípios de sistemas de bancos de dados distribuídos", tradução Vandenberg de Souza. Rio de Janeiro: Campus, 711 p.
- Pitoura, E. e Bhargava, B. (1995) "Maintaining Consistency of Data in Mobile Computing Environments". In proceedings of 15th IEEE International Conference on Distributed Computing Systems, June.
- Pitoura, E. e Bhargava, B. (1999) "Data Consistency in intermittently connected distributed systems". In IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 11, no. 6.
- Rocha, T. e Toledo, M. B. F. (2005) "Estudo de Modelos de Transação para o ambiente de Computação Móvel", Campinas: Instituto de Computação, Unicamp, 32 p., Relatório Técnico IC-05-25, Setembro.
- Rocha, T. e Toledo, M.B.F. (2007) "Mecanismos de Adaptação para Transações em Ambientes de Computação Móvel". In IEEE Latin America Transactions, vol.5, no.8.
- Russel, S. e Norvig, P. (2003) "Inteligência Artificial". Editora Campus.
- Serrano-Alvarado, P., Roncancio, C. e Adiba, M. (2004a) "A Survey of Mobile Transactions", Distributed and Parallel Databases, Kluwer Academic Publishers, 38 p.
- Serrano-Alvarado, P. (2004b) "Transactions Adaptables pour les Environments Mobiles." Université Joseph Fourier.
- Walborn, G. e Chrysanthis, P. (1995) "Supporting Semantics-Based Transaction Processing in Mobile Database Applications". Dans Symposium on Reliable Distributed Systems (SRDS), Bad Neuenahr, Germany, September.
- Walborn, G. e Chrysanthis, P. (1998) "Transaction Processing in PRO-MOTION". Proc. Of the Symposium on Applied Computing. ACM SAC V. 11, p.389-398.
- Yeo, L. e Zaslavsky, H. (1994) "A Submission of Transactions from Mobile Workstations in a Cooperative Multidatabase Processing Environment". Proceedings of the 14th International Conference on Distributed Computing Systems.