# Coerência de cache em sistemas cooperativos com apoio à mobilidade

Carla Diacui Medeiros Berkenbrock <sup>1</sup>
Celso Massaki Hirata <sup>2</sup>

Resumo: Os avanços da computação móvel em adição às novas formas de conectividade permitem a integração de dispositivos móveis em aplicações cooperativas. Entretanto, a maior parte das aplicações cooperativas síncronas correntes considera ambientes fortemente acoplados, onde a desconexão dos dispositivos móveis em geral não é levada em conta. Neste trabalho é definida a estratégia *Cache Coherence Scheme for Mobile Cooperative Work* (CCS-MoCW). Essa estratégia é baseada na disseminação de notificações de alteração, mescla características das abordagens *stateful* e *stateless*, interações síncronas e assíncronas, bloqueios e disponibiliza aspectos de percepção. A avaliação da estratégia CCS-MoCW foi conduzida por meio de experimentos controlados. Esses experimentos envolvem a elaboração de um estudo de caso. Os resultados dos experimentos mostram que a estratégia apresenta-se como solução para garantir a coerência de cache, o controle de concorrência e permitir a apresentação de aspectos de percepção em um ambiente cooperativo com apoio à mobilidade dos usuários.

Palavras-chave: Coerência de cache. Colaboração. CSCW. Mobilidade.

Abstract: The advance of mobile computing along with the new ways of connectivity allows the integration of mobile devices in cooperative applications. However, a great deal of current cooperative applications consider only strongly coupled environments where disconnection is not considered. In this paper we define a Cache Coherence Scheme for Mobile Cooperative Work (CCS-MoCW). This scheme is based on the dissemination of notifications. It also merges features of both stateless and stateful servers, synchronous and asynchronous interactions, blocking protocol, and provides aspects of awareness information. The evaluation of the CCS-MoCW scheme was conducted through controlled experiments. In these experiments, a prototype and a case study were developed. The results of the experiments indicate that the scheme ensures cache coherence, concurrency control and provides awareness information for mobile cooperative work.

Keywords: Cache coherence. Collaboration. CSCW. Mobility.

### 1 Introdução

O uso de ferramentas computacionais que permitam a cada participante cooperar na realização de uma determinada tarefa, independentemente de sua localização geográfica, pode reduzir o número de encontros necessários para completar os serviços, diminuindo, dessa forma, seu custo e tempo de entrega. Dentre os aspectos importantes

http://dx.doi.org/10.5335/rbca.2015.3931

<sup>&</sup>lt;sup>1</sup>Departamento de Ciência da Computação, Universidade do Estado de Santa Catarina - Udesc, Joinville - SC, Brasil.

<sup>{</sup>carla.berkenbrock@udesc.br}

<sup>&</sup>lt;sup>2</sup>Divisão de Ciência da Computação, Instituto Tecnológico de Aeronáutica - ITA, S.J. Campos - SP, Brasil. {hirata@ita.br}

para a realização de trabalhos compartilhados, destacam-se as informações sobre o usuário, o compartilhamento flexível de material de trabalho e do apoio à comunicação entre os membros dos grupos. Nesse contexto, o campo de pesquisa *Computer Supported Cooperative Work* (CSCW) fornece um entendimento de como as pessoas trabalham juntas, bem como de que forma os sistemas podem ser projetados para acomodar as suas práticas de trabalho [1].

Os dispositivos móveis permitem a troca de informações entre usuários que necessitam de mobilidade para obtenção de uma melhor eficiência no desenvolvimento de seus trabalhos. Entretanto, esses dispositivos introduzem dificuldades que não estão presentes em ambientes convencionais, i.e. ambientes onde os participantes têm localização fixa e cooperam a partir de computadores pessoais[2]. Dentre as dificuldades, podem-se citar o tamanho da tela dos dispositivos, problemas relacionados com a compatibilidade de software, a consistência de informações e o limitado poder de bateria.

Aplicações cooperativas necessitam de métodos apropriados para o controle de concorrência e sincronização[3]. As dificuldades aumentam quando essas aplicações envolvem ambientes de computação móvel e sem fio. Nesses ambientes, um membro de um grupo cooperativo, ao se desconectar da célula de comunicação sem fio, perde o contato com os demais participantes do grupo.

Neste trabalho, ressalta-se a importância do desenvolvimento de estratégias diretamente projetadas para trabalhar com coerência de cache em ambientes cooperativos. Isso porque as estratégias correntes oferecem um suporte limitado às atividades cooperativas. Por exemplo, quando mais de um usuário opera um documento, podese editar uma única informação de forma concorrente. Para evitar inconsistências, muitas das técnicas correntes invalidam a última alteração realizada. Em um ambiente cooperativo, esse comportamento poderá desmotivar os usuários e comprometer o andamento das atividades cooperativas.

Neste artigo, é definida uma estratégia para prover coerência de cache e elementos de percepção em um ambiente cooperativo com apoio à mobilidade dos usuários. De forma específica, busca-se uma estratégia que possibilite um balanceamento entre a obtenção de um resultado coerente e o desenvolvimento cooperativo de atividades por usuários portadores de dispositivos móveis. De modo a propiciar a avaliação da referida estratégia, são conduzidos experimentos controlados realizados por meio do desenvolvimento e da utilização de um estudo de caso desenvolvido em um cenário real. No estudo de caso, a análise da estratégia é realizada por meio da utilização de uma aplicação construída para fornecer suporte à estratégia, bem como para permitir a coleta de dados empíricos. Os experimentos realizados mostram que a estratégia possibilita coerência de cache e controle de concorrência, além de permitir a apresentação de aspectos de percepção em um ambiente cooperativo com apoio à mobilidade dos usuários.

Este artigo está organizado da seguinte forma. Na seção 2 são discutidos alguns trabalhos relacionados. A seção 3 descreve a estratégia *Cache Coherence Scheme for Mobile Cooperative Work* (CCS-MoCW). A seção 4 descreve o estudo de caso realizado para analisar a viabilidade da estratégia CCS-MoCW em um cenário real. A seção 5 apresenta a análise dos resultados. Por fim, na seção 6 são apresentadas as contribuições obtidas e as direções futuras desta pesquisa.

#### 2 Trabalhos relacionados

Barbara e Imieliski[4] estão entre os primeiros a utilizar NIs para tratar o problema de coerência de cache em ambiente sem fio. Eles propuseram as estratégias *Broadcasting Timestamp* (TS) e *Amnesic Terminals* (AT).

Essas estratégias estão focadas na invalidação de cache onde o servidor envia uma NI periodicamente para os clientes móveis. Os clientes utilizam essa notificação para invalidar sua cache. As duas estratégias diferenciam-se em relação ao conteúdo da NI. Se o tempo de desconexão exceder um valor predeterminado, os clientes são forçados a descartar toda a cache após a reconexão. Essas estratégias não permitem interação entre os usuários. Ainda, é possível que vários usuários alterem um mesmo objeto, ocasionando inconsistências. Adicionalmente, as NIs não incluem o novo conteúdo dos objetos após as alterações. Se for realizada alguma alteração em objetos que estão armazenados na cache, esses objetos serão excluídos dos clientes, ainda que o objeto esteja em alteração.

Chung e Cho[5] propõem uma estratégia para coerência de cache chamada *Cache Coherency Schema with Incremental Update Propagation* (CCS-IUP). Essa estratégia diferencia-se das abordagens propostas por Barbara e Imieliski principalmente porque evita a invalidação da cache mediante a propagação das novas informações que foram alteradas ou inseridas. Embora diminua o número de invalidações, essa abordagem também não foi projetada para possibilitar acesso concorrente com resultados consistentes.

Wang et al[6] propõem uma estratégia de coerência de cache chamada *Scalable Asynchronous Cache Consistency Scheme* (SACCS). Nela são herdadas as características positivas das abordagens *stateless* e *stateful*. SACCS fornece consistência fraca em ambientes móveis e sem fio pelo uso de notificações de invalidação. Essa abordagem permite estimar o tempo de vida de cada registro armazenado na cache. Contudo, ela foi projetada apenas para operações de leitura na cache.

Safa, Hassan e Nahhas[7] propõem a estratégia *Selective Adaptive Sorted* (SAS). O objetivo dessa estratégia é tratar o problema da falsa invalidação e melhorar a utilização de banda, reduzindo o tamanho das notificações e do número de pedidos de *uplink* enviados para o servidor.

As limitações apresentadas restringem os cenários de aplicação onde se pode fazer uso das estratégias definidas em [4], [5], [6] e [7]. Elas podem ser utilizadas em ambientes onde as alterações não são frequentes, ou em situações onde a ação de um usuário não interfira diretamente no conteúdo dos demais, como, por exemplo, na consulta de informações de negócios, vendas recentes, dados de estoque, temperatura ou notícias. Nessas situações, o usuário pode se manter informado sobre os dados de interesse e solicitar uma atualização periódica desses dados. Adicionalmente, essas estratégias não foram projetadas para sistemas colaborativos. Geralmente, os trabalhos relacionados com a coerência de cache em ambientes móveis preocupam-se com a redução da quantidade de mensagens trocadas e a eficiência da coerência, no entanto, não levam em consideração a percepção e o desempenho do trabalho cooperativo.

Papadopoulos [8] investiga oportunidades para melhorar a percepção em ambientes móveis. O autor apresenta um sistema colaborativo síncrono e define um protocolo que propõe um balanceamento entre consumo de energia e tempo de notificação. O objetivo do autor é fornecer percepção para colaboradores móveis de maneira eficiente em termos de tempo e consumo de energia. As informações de percepção são apresentadas seguindo um modelo relaxado de interface *What You See Is What I See* (WYSIWIS). O ambiente considerado permite ao usuário a mudança de localização, cruzando células. A estação responsável pela cobertura da célula em que o usuário se localiza possibilita a continuação da comunicação. Entretanto, a mudança de localização pode ocasionar desconexões. Enquanto o usuário permanecer desconectado, seu dispositivo móvel entra no modo adormecido para salvar energia. Com isso, o usuário deixa de realizar ações durante o período de desconexão.

Dimokas et al. (2011) [9] propõem as estratégias *Power Comunity Index Cooperative Caching* (PCICC) e *scaled version of the PCICC* (scaPCICC) para promover um novo protocolo de cache cooperativa para sensores de redes sem fio. A estratégia foi inspirada na ideia do índice de importância do nodo, porém, dois quesitos

foram adicionados: rede sem fio e cache cooperativa. A estratégia *u-power community index* (u-PCI) propõe novas métricas para escolha do nodo central. A estratégia *a scaled version of the u-PCI* (scaPCI) procura otimizar a escolha do nodo central quando há um número grande de nodos. A ideia principal das estratégias PCICC e scaPCICC é escolher o melhor nodo central, capaz de aperfeiçoar a comunicação entre os demais nodos da rede.

# 3 Descrição da estratégia CCS-MoCW

O problema da coerência de cache é tratado neste trabalho, pela perspectiva da arquitetura híbrida semirreplicada, onde cada unidade móvel (UM) executa uma cópia do programa de aplicação e apresenta uma réplica das informações compartilhadas pelos participantes do sistema colaborativo. As UMs podem alterar as suas réplicas locais (cache), entretanto, essas alterações apenas poderão ser efetivadas pela estação fixa (EF). A EF recebe as requisições das UMs, e, assim, que estas são atendidas, as alterações são propagadas para as UMs conectadas ao grupo cooperativo.

A estratégia CCS-MoCW é composta por dois procedimentos principais: (*i*) a coerência de cache e percepção do servidor, implementada na EF; e (*ii*) a coerência de cache e percepção do cliente, implementada nas UMs. Esses procedimentos são apresentados, respectivamente, nas Figuras 3 e 3. A Tabela 1 sumariza os parâmetros usados na estratégia CCS-MoCW.

Tabela 1: Parâmetros na estratégia CCS-MoCW

Parâmetros	Descrição
d	Tempo de desconexão da unidade móvel
HA	Histórico de alterações
iB	Intervalo de <i>broadcast</i>
id	Identificador único de registro
LB	Lista de bloqueios
LP	Lista de pendências
NA	Notificação de alteração
P	Período máximo de desconexão
$RT_{ANT}$	Rótulo de tempo anterior da estação fixa
$RT_{EF}$	Rótulo de tempo corrente da estação fixa
$RT_{UM}$	Rótulo de tempo da unidade móvel
w	Janela de desconexão

Ao iniciar suas atividades, a EF inicia as variáveis responsáveis pela coerência de cache e percepção com valores preestabelecidos no sistema. O rótulo de tempo da EF  $(RT_{EF})$  é iniciado com o valor zero, e esse valor é atualizado pela EF a cada envio de notificação de alteração (NA).  $RT_{ANT}$  é utilizado para identificar se a UM perdeu alguma mensagem de NA enviada previamente.  $RT_{EF}$  é utilizado para que as que as UMs atualizem o seu rótulo de tempo  $(RT_{UM})$ . A NA também contém informações sobre os dados que foram alterados durante o último intervalo de broadcast e sobre as ações realizadas pelos usuários nesse intervalo. Essas informações ajudam os usuários a terem percepção sobre as atividades realizadas durante a cooperação.

Na estratégia, usam-se bloqueios para permitir que as operações sejam realizadas de maneira concorrente e, ainda assim, o resultado das operações seja equivalente a operações realizadas de forma serial. Além disso, usam-se *timeouts* para detectar possíveis desconexões e evitar que UMs desconectadas, segurando bloqueios, conduzam o sistema a situações de impasse (*deadlock*).

Para cada registro no grupo cooperativo é atribuído um identificador único de registro (*id*). Inicialmente, as UMs têm apenas acesso à leitura dos dados que estão armazenados em cache. A UM apenas tem acesso à escrita de um registro quando consegue bloqueá-lo.

Figura 1: Procedimento para a coerência de cache e percepção da EF

```
Para cada NA a ser enviada pela EF
   RTant := RTef
   RTef := timestamp corrente
   Adiciona na NA RTant, RTef e os dados alterados após RTant
   Para cada NA enviada
      Para cada UM no grupo cooperativo
         Se recebe ACK
           d := 0 (Registra que a UM está conectada)
         Senão
            Incrementa d
            Se (d > P)
              Altera o bloqueio da UM para vulnerável na LB
         Se recebe SYNC
            Se (RTef - RTum <= w)
               Envia ANSWER com dados para a nova sincronização
            Senão
               Envia ANSWER com dados para invalidação
Para cada mensagem recebida pela EF
   Se mensagem = JOIN
      Registra entrada da UM no HA
      Envia ANSWER com réplica dos dados do grupo
   Se mensagem = LOCK
      Se registro não está bloqueado ou possui bloqueio vulnerável
         Registra bloqueio da UM no HA
         Registra bloqueio da UM na LB
         Se registro foi alterado após RTef
           Envia ANSWER de confirmação do bloqueio e a última versão do registro
         Senão
           Envia ANSWER de confirmação do bloqueio
      Senão
         Registra solicitação de bloqueio da UM na LP
         Envia ANSWER informando que o registro já está bloqueado
   Se mensagem = UPDATE
      Se (RTum = RTef)
         Registra alterações no HA
         Registra desbloqueio desse registro na LB
         Envia ANSWER de confirmação da alteração
      Se (RTef - RTum <= w)
         Se ((d < P) ou (d > P e UM continua segurando o bloqueio))
            Registra alterações no HA
            Registra desbloqueio desse registro na LB
            Envia ANSWER com dados para a nova sincronização
         Senão
            Envia ANSWER informando que alteração não pode ser realizada,
            em conjunto com dados para a nova sincronização
      Se (RTef - RTum > w)
         Envia ANSWER com dados para invalidação
      Envia ID-FREE para UMs interessadas contidas na LP
      Retira UMs da LP
   Se mensagem = CANCEL
      Registra desbloqueio desse registro na LB
      Envia ANSWER de confirmação do cancelamento
      Envia ID-FREE para UMs interessadas contidas na LP
      Retira UM da LP
```

Figura 2: Procedimento para a coerência de cache e percepção das UMs

```
Para cada mensagem recebida pela UM
  Se mensagem = NA
        Se (RTant = RTum)
           Para cada registro na NA
              Substitui os dados em cache pelos dados contidos na NA
         Senão
           Envia SYNC
            Se recebe ANSWER com dados para a nova sincronização
              Para cada registro recebido
                 Substitui os dados em cache pelos dados recebidos
              Se recebe ANSWER com dados para invalidação
                  Exclui todos os dados da cache
                 Armazena em cache a nova réplica dos dados do grupo recebida
         RTum := RTef
        Envia ACK
   Se mensagem = ID-FREE
     Informa UM sobre desbloqueio
Para cada mensagem enviada pela UM
  Se mensagem = JOIN
     Se recebe ANSWER com réplica dos dados do grupo
        Armazena em cache a réplica dos dados do grupo recebida
        RTum := RTef
  Se mensagem = LOCK
     Se recebe ANSWER de confirmação de bloqueio liberado
        UM começa a alterar o registro
   Se mensagem = UPDATE
     Se recebe ANSWER com dados para a nova sincronização
        Para cada registro recebido
           Substitui os dados em cache pelos dados recebidos
         Se recebe ANSWER com dados para invalidação
           Exclui todos os dados da cache
           Armazena em cache a nova réplica dos dados do grupo
   Se mensagem = CANCEL
     Informa a resposta da EF para UM
```

Um bloqueio vulnerável indica que o usuário, segurando esse bloqueio, está desconectado por um período de tempo superior ao período máximo de desconexão preestabelecido pelo sistema. Nesse caso, para evitar situações de impasse, o sistema permite que outros usuários tenham acesso ao registro que possui o bloqueio vulnerável.

Quando a EF recebe uma mensagem para bloquear um determinado registro identificado pelo *id*, ela verifica se nenhuma outra UM está segurando bloqueio para esse registro ou se existe bloqueio vulnerável para esse registro. Em ambos os casos, o bloqueio é concedido. Antes de enviar a mensagem de confirmação de bloqueio, a EF verifica se os dados do registro solicitado foram alterados após o último envio da *NA*. Isso porque, como a UM solicitante tinha apenas acesso à leitura do registro, outra UM pode ter realizado alterações no registro que ainda não estão no cache da UM solicitante do bloqueio. Assim, havendo alteração, a EF envia a última versão do registro para que UM solicitante do bloqueio atualize sua cache e, com isso, realize as novas alterações considerando os dados correntes do registro. Portanto, não há possibilidade de que uma UM sobreponha e invalide uma informação já efetivada.

As mensagens trocadas entre EF e UMs são apresentadas na Tabela 2.  $UM_{id}$  é o identificador único da UM e nv é o novo valor do registro alterado pela UM. O par [id, vl] contém os identificadores dos registros do grupo cooperativo em conjunto com seus dados. Por fim, o parâmetro rs contém o resultado da solicitação realizada pela UM para a EF, onde rs = 1 indica que a operação foi realizada com sucesso e rs = 0 indica problemas na operação.

Tabela 2: Mensagens para comunicação na estratégia CCS-MoCW

	<u> </u>			
Mensagem	Descrição	Conteúdo	Envia	Recebe
JOIN	Solicitação de entrada no grupo cooperativo	$UM_{id}$	UM	EF
LOCK	Solicitação de bloqueio	$UM_{id}$ , $id$	UM	EF
UPDATE	Solicitação de confirmação de alteração	$UM_{id}, RT_{UM}, id, nv$	UM	EF
CANCEL	Solicitação de cancelamento de bloqueio	$UM_{id}$ , $id$	UM	EF
SYNC	Solicitação de nova sincronização	$UM_{id}, RT_{UM}, id$	UM	EF
NA	Notificação de alteração	$RT_{ANT}, RT_{EF}, [id, vl]$	EF	UMs
ID-FREE	Aviso sobre desbloqueio de um registro	id	EF	UMs
ACK	Aviso sobre recebimento de NA	$UM_{id}$	UM	EF
ANSWER	Resposta da EF para a solicitação da UM	$rs, RT_{EF}, [id, vl]$	EF	UMs

Na resposta da EF para as solicitações das UMs, o valor do par [id, vl] varia de acordo com o tipo de solicitação recebida. Caso seja de resposta da EF para uma solicitação de entrada no grupo cooperativo, [id, vl] contém os identificadores e respectivos dados de todos os registros do grupo cooperativo. Caso seja de resposta da EF para uma solicitação de bloqueio, [id, vl] pode estar vazio ou conter o valor da última versão do registro desejado. Caso seja de resposta da EF para uma solicitação de confirmação de alteração, [id, vl] pode estar vazio, ou conter dados alterados após  $RT_{UM}$ . Caso seja de resposta da EF para uma solicitação de cancelamento de bloqueio, [id, vl] fica vazio. Caso a mensagem seja de resposta da EF indicando nova sincronização: [id, vl] contém os dados alterados após  $RT_{UM}$ . Caso a mensagem seja de resposta da EF indicando invalidação, [id, vl] contém todos os dados correntes do grupo.

## 4 Estudo de caso

O ambiente de desenvolvimento e de experimento é composto por quatro unidades móveis, um servidor sob Linux Ubuntu e uma infraestrutura de rede local sem fio segundo o padrão IEEE 802.11. A comunicação entre as UMs e a EF é feita com soquetes TCP/IP. As notificações de alteração são transmitidas a cada 60 segundos (*iB*), e a janela de *broadcast* é de 600 segundos (*w*). O período máximo de desconexão (*P*) considera três intervalos de *broadcast* (180 segundos).

A perspectiva lúdica aliada à mobilidade é uma alternativa na busca de conhecimento[10]. Neste trabalho, foi desenvolvido o jogo colaborativo chamado "Warming up the Brain", a fim de demonstrar que a estratégia fornece o comportamento correto para aplicações cooperativas.

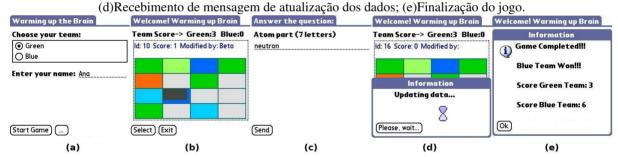
O *Warming up the Brain* é um jogo de perguntas e respostas desenvolvido para ser jogado individualmente, concorrentemente ou cooperativamente. Contudo, considerando que o objetivo deste trabalho é analisar a estratégia CCS-MoCW em ambientes de cooperação móvel, apenas é analisada a opção cooperativa.

O objetivo do jogador é conseguir o maior número de pontos, respondendo corretamente a um conjunto de questões que lhe são apresentadas. Os jogadores podem se dividir em duas equipes, sendo cada equipe representada por uma cor – azul ou verde. Antes de entrar no grupo cooperativo, o usuário deve selecionar a equipe na qual deseja jogar, como apresentado na Figura 3(a).

As perguntas do jogo são representadas por meio de uma matriz bidimensional. Cada posição na matriz corresponde a uma pergunta, mantida oculta dos jogadores antes da seleção. O jogador deve posicionar o cursor sobre uma posição e clicar na opção *Select* para selecioná-la, conforme apresentado na Figura 3(b). Se a questão estiver disponível, o jogador tem acesso à pergunta e pode respondê-la, conforme apresentado na Figura 3(c). Não existe sequência para jogar; qualquer um de cada grupo pode fazer uma jogada a qualquer momento.

Os jogadores recebem uma notificação de alteração (NA) a cada 60 segundos. A Figura 3(d) apresenta a mensagem exibida na tela do participante, cada vez que ele recebe uma NA. A notificação de alteração contém as informações sobre as alterações realizadas pelos demais jogadores e é utilizada para sincronizar os dados do jogador com os dados dos demais participantes do jogo. O jogo acaba quando todas as perguntas forem respondidas corretamente, ou quando o tempo disponível para o jogo acabar, como mostrado na Figura 3(e).

Figura 3: (a) Tela inicial; (b) Matriz de perguntas; (c) Questão selecionada e respectiva resposta do jogador;



As cores em cada posição na matriz revelam o estado das perguntas. Existem cinco diferentes variações de cores, a saber:

- cinza claro a pergunta ainda não foi respondida. Uma resposta correta contabiliza um ponto para a equipe. Por exemplo, se a equipe azul responder corretamente à questão, a posição na matriz é colorida com azulescuro. Caso a resposta correta tenha sido fornecida pela equipe verde, a posição na matriz é colorida com verde-escuro. Em caso de resposta errada, decrementa-se um ponto da equipe, e a posição correspondente à pergunta é colorida com azul-claro, se a resposta foi dada pela equipe azul, ou com a cor verde-claro, se a resposta foi dada pela equipe verde;
- azul-escuro/verde-escuro a equipe azul/verde acertou a resposta dessa questão. A questão não pode mais ser respondida;

- azul-claro/verde-claro a equipe azul/verde errou a resposta dessa questão. Nesse momento, a equipe que errou a questão recebe um ponto negativo para essa questão. Caso algum jogador, de qualquer equipe, deseje respondê-la, ele pode errar a resposta sem perder pontos. Caso a nova resposta esteja correta, se essa resposta foi dada por um membro da equipe que deu a resposta errada, o ponto negativo é anulado, porém, não são contabilizados pontos positivos. Se um membro da equipe adversária responder corretamente à pergunta, essa equipe recebe dois pontos positivos e é decrementado mais um ponto da equipe que deu a resposta errada, além do ponto negativo por ter dado a primeira resposta errada, o que totaliza dois pontos negativos; e
- vermelho a pergunta está sendo respondida por algum jogador, de forma que o acesso à posição está bloqueado a todos os jogadores.

A pontuação do jogo é dada pela soma de todos os pontos obtidos pelos membros de cada equipe.

#### 4.1 Estrutura do experimento controlado

O experimento controlado foi realizado por doze participantes, sete homens e cinco mulheres, com idades entre 25 e 45 anos (idade média = 31,5 anos e  $\sigma$ =6). A maioria desses participantes (83,33%) compreende estudantes de pós-graduação em Computação. A maioria dos participantes (58,33%) também indicou nunca ter participado de jogos em rede com múltiplos membros.

Os participantes foram divididos em três grupos compostos de quatro componentes. No contexto do jogo cooperativo, para que seja possível analisar a cooperação é importante ter um grupo de ao menos quatro jogadores, distribuídos em duas equipes.

Conforme mostrado na Figura 4, no ambiente do experimento os participantes estavam próximos uns dos outros (*co-located*). Entretanto, apesar de estarem em uma mesma sala, cada um usando o seu dispositivo móvel, a orientação era de que qualquer comunicação fosse feita apenas por meio dos dispositivos móveis. Cada participante recebeu um nome fictício para participar no jogo. Sendo assim, ninguém conhecia o seu parceiro nem os seus adversários.

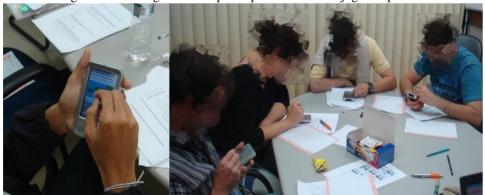


Figura 4: Dois flagrantes dos participantes durante o jogo cooperativo

Os instrumentos de coleta dos dados para a avaliação foram os seguintes: (i) observações feitas pelos mediadores da sessão durante os experimentos; (ii) logs registrados na EF e nas UMs contendo informações sobre as

trocas de mensagens realizadas durante a cooperação; e (iii) questionários preenchidos pelos participantes durante o experimento.

Os mediadores da sessão foram responsáveis por fornecer as instruções sobre o experimento aos participantes. Antes de iniciar as tarefas monitoradas, cada participante recebeu um *Documento de Boas vindas*, contendo algumas informações sobre o jogo cooperativo e um *Questionário Pré-Experimento*, buscando identificar o perfil do participante e a sua experiência anterior com dispositivos móveis. Em seguida, realizou-se uma apresentação do dispositivo empregado nos experimentos e promoveu-se um *Experimento Teste*, buscando familiarizar o participante com as características tanto do dispositivo móvel quanto do jogo cooperativo.

O *Experimento Monitorado* envolveu o uso do jogo cooperativo até que se obtivesse uma equipe vencedora, ou até que o tempo previsto para o jogo, 30 minutos, expirasse. Após cada experimento monitorado, aplicou-se o *Questionário Pós-Experimento*, com o qual se busca identificar elementos de percepção, coerência, compreensão das atividades e desempenho obtidos pelos participantes. Nesse documento, também se solicitava ao participante a indicação de sugestões para realização de levantamento das necessidades do usuário.

#### 5 Resultados coletados

O desempenho da estratégia CCS-MoCW no cenário do jogo cooperativo foi analisado de acordo com os parâmetros sumarizados na Tabela 3.

Tabela 3: Parâmetros analisados na estratégia CCS-MoCW

Tubbla 5. Turumburbs anamsados na estrategia e es 1410 e 44			
Descrição	Valor		
Taxa de solicitações para a EF por minuto	15,39 (σ=1,35)		
Tempo médio de desconexão por UM	35,29s (σ=4,53s)		
Número de alteração da cache por minuto	5,99 (σ=0,25)		
Número médio de desconexões por grupo	7,33 (σ=3,21)		
Número médio de invalidações por grupo	3 (σ=3)		
Porcentagem de mensagens perdidas	3,74%		
Porcentagem de coerência de cache	100%		
Porcentagem de consistência	99,36%		
Porcentagem de identificação de elementos de percepção	39%		

Durante os experimentos, cada grupo realizou, em média, 15,39 solicitações ( $\sigma$ =1,35) por minuto para a EF. Essas solicitações incluem todas as mensagens enviadas pelas UMs para a EF, conforme apresentado na Tabela 2. A cache de cada UM foi alterada, em média, 5,99 vezes ( $\sigma$ =0,25) a cada minuto. Essas alterações na cache não somente ajudam a manter a coerência da cache, como também informam os usuários sobre as mudanças no estado das perguntas (registros).

Os experimentos realizados consideram três situações de cooperação: (i) UM permanece conectada durante todo o processo de cooperação, (ii) UM se desconecta por um período curto de tempo e (iii) UM permanece desconectada por um longo período.

Em situações nas quais a UM permanece conectada durante todo o processo de cooperação (situação i), a preocupação maior da EF é a de garantir o controle de concorrência. Com isso, a EF é responsável por resolver os conflitos entre as operações simultâneas realizadas pelos participantes. Enquanto um participante está respondendo a uma determinada pergunta, os demais participantes não podem ter acesso a essa pergunta. Um novo acesso à pergunta apenas será liberado quando a operação em andamento for anulada ou efetivada. Contudo, nos experimentos, cada UM teve ao menos uma desconexão. Em média, ocorreram 7,33 desconexões ( $\sigma$ =3,21) em

cada grupo. Além disso, considerando que 3,74% das mensagens trocadas durante a cooperação não chegaram ao seu destino, ainda que as UMs permaneçam conectadas, existe o risco de perda de mensagens. Assim, além do controle de concorrência, a EF precisou se preocupar em manter a coerência de cache em situações de desconexão durante todas as sessões cooperativas.

As situações em que a UM permanece desconectada por um curto período de tempo (situação ii) são as mais complexas de serem resolvidas pela EF. Nos experimentos, considera-se como período curto as desconexões de duração inferior ou igual a 600 segundos. Em média, cada grupo teve 7,33 desconexões. O tempo médio de desconexão por UM foi de 35,29 segundos ( $\sigma$ =4,53). Durante os experimentos, 59,10% das desconexões ocorridas tiveram tempo de desconexão menor do que 180 segundos. Esse foi o caso mais comum de desconexão. Se o tempo de desconexão é maior do que 180 segundos e o participante continua segurando o bloqueio, isso significa que nenhum outro participante teve interesse em acessar a pergunta que estava sendo respondida pelo usuário desconectado. Contudo, esse caso não ocorreu durante os experimentos. Uma justificativa para esse acontecimento é que os usuários tentavam continuamente acessar as perguntas, ainda que o estado da pergunta indicasse ocupado. Além disso, o tempo de acesso em cada pergunta era pequeno (média de 26,33 segundos). Dessa maneira, assim que o registro tornava-se vulnerável, algum usuário já conseguia bloqueá-lo. Nos casos em que o tempo de desconexão é maior do que 180 segundos e o participante perdeu o direito de consolidar a sua resposta, isso significa que algum outro participante teve interesse em alterar a pergunta que estava sendo respondida pelo usuário desconectado. Dessa maneira, a EF forneceu o direito de resposta ao outro usuário. Com isso, após a reconexão, a resposta do usuário que permaneceu desconectado não poderá ser consolidada na EF. Assim, evita-se que informações sejam sobrescritas. Esse caso aconteceu em 40,90% das desconexões.

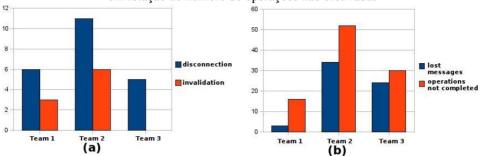
No experimento, a EF registra em um histórico todas as alterações que ocorreram no grupo cooperativo nos últimos 600 segundos. Assim, se a UM permaneceu desconectada por um período superior a esse tempo (situação *iii*), isso significa que podem ter ocorrido alterações no sistema colaborativo que já não estão armazenadas no histórico. Dessa forma, a EF não tem capacidade de verificar a coerência de cache. Sendo assim, a EF invalida as informações da UM reconectada e envia as informações correntes do grupo. Em cada grupo cooperativo ocorreram em média três invalidações de cache. A aplicação desenvolvida para o experimento ainda não possibilita que o participante tenha percepção sobre a quantidade de tempo em que ele se encontra desconectado. Esse tipo de percepção de lembrança (*reminding awareness*) pode ajudar o usuário a ter consciência a respeito do risco de perda dos dados que estão sendo alterados enquanto ele permanece desconectado. Contudo, considerando que, em média, cada grupo cooperativo recebeu 280,33 mensagens da EF, as invalidações representaram apenas 1,07% das mensagens recebidas.

A observação dos registros de *log* permitiu verificar a taxa de mensagens perdidas. Durante a cooperação, foram trocadas 1.633 mensagens entre a EF e as UMs. Dessas mensagens, 3,74% não chegaram ao seu destino. Devido à perda de mensagens, 19,68 % das operações iniciadas pelos participantes não puderam ser efetivadas.

A Figura 5(a) mostra que os grupos que tiveram mais desconexões também tiveram maior número de invalidações. Na Figura 5(b), verifica-se que os grupos que perderam mais mensagens tiveram maior número de operações não efetivadas.

Apesar de a taxa de perda de mensagem ser significativa, a estratégia CCS-MoCW garantiu 100% da coerência de cache no sistema. No final da colaboração, a cache de todas as unidades móveis estava coerente com os dados da EF. Contudo, o sistema apresentou algumas inconsistências. Ocorreram duas situações onde o controle de concorrência do sistema apresentou problemas. Com isso, o sistema obteve 99,36% de consistência.

Figura 5: (a)Número de desconexões em relação ao número de invalidações; (b)Número de mensagens perdidas em relação ao número de operações não efetivadas



A estratégia possibilitou, ainda, a apresentação de elementos de percepção. O questionário aplicado após a realização dos experimentos foi utilizado para coletar dados sobre os elementos de percepção identificados pelos usuários. Com base nas respostas desse questionário, tem-se que a taxa de identificação de elementos de percepção foi de 39%. Os problemas encontrados na análise dos requisitos de percepção não demonstram, necessariamente, limitações da estratégia em fornecer informações de percepção, mas sim em como apresentar essas informações no dispositivo de maneira que elas sejam percebidas e entendidas pelos usuários.

De forma similar, os retardos nos tempos de resposta não representam, necessariamente, problemas na estratégia, mas sim no projeto da aplicação. Por exemplo, no jogo cooperativo, as perguntas não estão inicialmente armazenadas na cache do dispositivo. Assim, cada vez que o usuário tem acesso a uma pergunta, a EF precisa enviar a pergunta para a UM. Esse excesso de comunicação poderia ser evitado, caso todas as informações relacionadas com as perguntas e respostas já estivessem de antemão armazenadas na cache do dispositivos. Assim, apenas mensagens de controle seriam transmitidas, avisando sobre a permissão ou não do acesso à pergunta.

As regras relacionadas com o cancelamento de bloqueio e com o aviso sobre desbloqueio de registros não se adéquam ao jogo cooperativo, portanto, essas regras não foram implementadas na aplicação desenvolvida. A partir do momento em que o usuário consegue bloquear uma questão, ele acessa a pergunta a ela correspondente. Portanto, caso o usuário desconhecesse a resposta dessa questão, se a opção de cancelamento de bloqueio estivesse acessível, ele poderia cancelar o bloqueio para evitar perder pontos.

No jogo cooperativo, o estado das perguntas é alterado frequentemente (cerca de seis alterações por minuto). Assim, para evitar interrupções da atenção do usuário, devido à quantidade excessiva de mensagens sobre desbloqueios de registros, as regras relacionadas com o aviso sobre desbloqueios não foram implementadas. Além disso, as cores em cada posição na matriz auxiliam o usuário a ter percepção sobre a mudança de estado das perguntas.

#### 6 Conclusões e trabalhos futuros

Neste trabalho, foi definida uma estratégia para garantir a manutenção da coerência de cache e fornecer elementos de percepção para ambientes colaborativos móveis. A avaliação da estratégia foi conduzida por meio de experimentos controlados. Os experimentos realizados mostram que a estratégia CCS-MoCW atende aos requisitos especificados, ou seja, ela possibilita coerência de cache, controle de concorrência e permite a apresentação de aspectos de percepção em um ambiente cooperativo com apoio à mobilidade dos usuários.

A estratégia CCS-MoCW se mostrou eficaz no sentido de manter o resultado do trabalho coerente, porém, a sua utilização restringe a liberdade do usuário. Isso porque o desenvolvimento das atividades precisa seguir algumas regras estabelecidas pela estratégia, o que torna o uso do ambiente menos amigável.

Neste trabalho, constatou-se, ainda, que o projeto e a análise da usabilidade do ambiente onde a referida estratégia será utilizada se tornam quase tão importantes quanto o uso da estratégia, pois é por meio do uso da aplicação que o usuário consegue desenvolver suas atividades, de modo a alcançar o seu objetivo computacional.

Alguns problemas identificados neste trabalho estão em aberto e serão endereçados futuramente. Dentre as sugestões para trabalhos futuros, destacam-se: (i) trabalhar com a noção de contexto, de modo a adaptar as aplicações para as condições e limitações de cada usuário; (ii) analisar a aplicabilidade de jogos cooperativos em dispositivos móveis para a área de educação; e (iii) definir métricas e requisitos de usabilidade, a fim de aperfeiçoar e melhorar o projeto e a usabilidade de aplicações cooperativas móveis.

### Agradecimentos

Ao apoio da FITEJ.

### Referências

- [1] M. S. Ackerman, Everything you need to know about collaboration and collaboration software *Proceedings* of the 19th international conference on Software engineering (ACM Press, Boston, 1997).
- [2] M. J. Mantau, C. D. M. Berkenbrock, G. R. Berkenbrock. Visualization and Filtering Awareness Information in Mobile Groupwares: an Action Research Approach (CSCWD 2014).
- [3] C. D. M. Berkenbrock, C. M. Hirata, Supporting Cache Coherence in Mobile Cooperative Systems. In: The 7th IEEE International Symposium on Network Computing and Applications (NCA08), p. 240–243. (2008)
- [4] D. Barbara, T. Imieliski, Sleepers and workaholics: caching strategies in mobile environments *Proceedings* of the 1994 ACM SIGMOD international conference on Management of data, 2nd edn. (1994).
- [5] H. Chung, H. Cho, Data caching with incremental update propagation in mobile computing environments. Australian Computer Journal, v. 30, n. 2, p. 77–86, (1998).
- [6] Z. Wang and S. K. Das, H. C. and M. Kumar, A Scalable Asynchronous Cache Consistency Scheme (SACCS) for Mobile Environments, in *IEEE transactions on parallel and distributed systems* (2004).
- [7] H. Safa, H. Artail, M. Nahhas, A cache invalidation strategy for mobile networks. In: Journal of Network and Computer Applications, p. 168–182. ISSN 1084-8045, (2010).
- [8] C. Papadopoulos, Improving Awareness in Mobile CSCW, in *IEEE Transactions on Mobile Computing*, (IEEE Computer Society, 2006).
- [9] N. Dimokas, D. Katsaros, L. Tassiulas, Y. Manolopoulos, High Performance, Low Complexity Cooperative Caching for Wireless Sensor Networks, in *Wirel. Netw.* (Springer-Verlag New York, Inc. 2011)
- [10] A.Mühlbeier, P.Mozzaquatro, L.Oliveira, T.Monteiro, V.Lopes, eNIGMA e M-Learning: jogo educativo trabalhando o raciocínio lógico através de dispositivos móveis, Revista Brasileira de Computação Aplicada (2012)