Concretização do suporte à Comunicação Grupo no sistema Appia

Alexandre Jorge Matos Pinto

2006

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

Appia – suporte à Comunicação em Grupo

- O Appia é distribuído com um conjunto de micro-protocolos para Comunicação em Grupo.
  - package `org.continuent.appia.protocols.group`
- Inspirado no Ensemble
- Funciona sobre camadas que oferecem comunicação ponto-a-ponto fiável.
- Decomposto em vários protocolos.
- Oferece ***Sincronia na Vista (view sync)***
- Notas:
  - Não existe mecanismo explicito que permita juntar um elemento ao grupo
    - O elemento que pretende entrar no grupo cria uma vista em que é o único membro e espera que o mecanismo de união de vistas junte a sua vista à vista *principal.*

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

Sincronia na Vista *(view sync)*

- Informação sobre a filiação do grupo entregue em vistas *(view)*.
  - Classe `ViewState`
- Vistas entregues pela mesma ordem em todos os membros activos.
- Mensagens podem ser associadas com a vista em que foram enviadas.
  - Se *m* é enviada na vista *v* então *m* é entregue em todos os membros activos na vista *v*.
  - Implica que outros podem ter que retransmitir mensagens recebidas:
    - **v** = { **A** , **B** , **C** , **D** }
    - **A** envia **m**
    - **A** falha
    - **m** recebida/entregue por **B** e **C**, mas não por **D**
    - **B**, ou **C**, retransmite **m** para **D**
    - **v'** = { **B** , **C** , **D** }

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

Comunicação em Grupo

- Classes comums importantes (package `org.continuent.appia.protocols.group`):
  - `ViewState`
    - Informação global da vista → igual em todos os membros.
      - `String version` → versão, actualmente é ignorado
      - `Group group` → identificação do grupo
      - `ViewID id` → identificação da vista actual
      - `ViewID[] previous` → identificação das vistas anteriores
      - `Endpt[] view` → membros da vista
      - `InetSocketAddress[] addresses` → endereços dos membros
  - `LocalState`
    - Informação local da vista → diferente em cada membro
      - `boolean[] failed` → membros falhados
      - `int my_rank` → a minha ordem na vista
      - `int coord` → a ordem na vista do coordenador
      - `boolean am_coord` → sou o coordenador ?
  - `Endpt`
    - Identificador único de cada membro.
  - `Group`
    - Identificador único do grupo.
  - `ViewID`
    - Identificador único da vista, composto pelo:
      - Identificador único do coordenador inicial;
      - Relógio lógico da vista actual → nº de vistas anteriores + 1

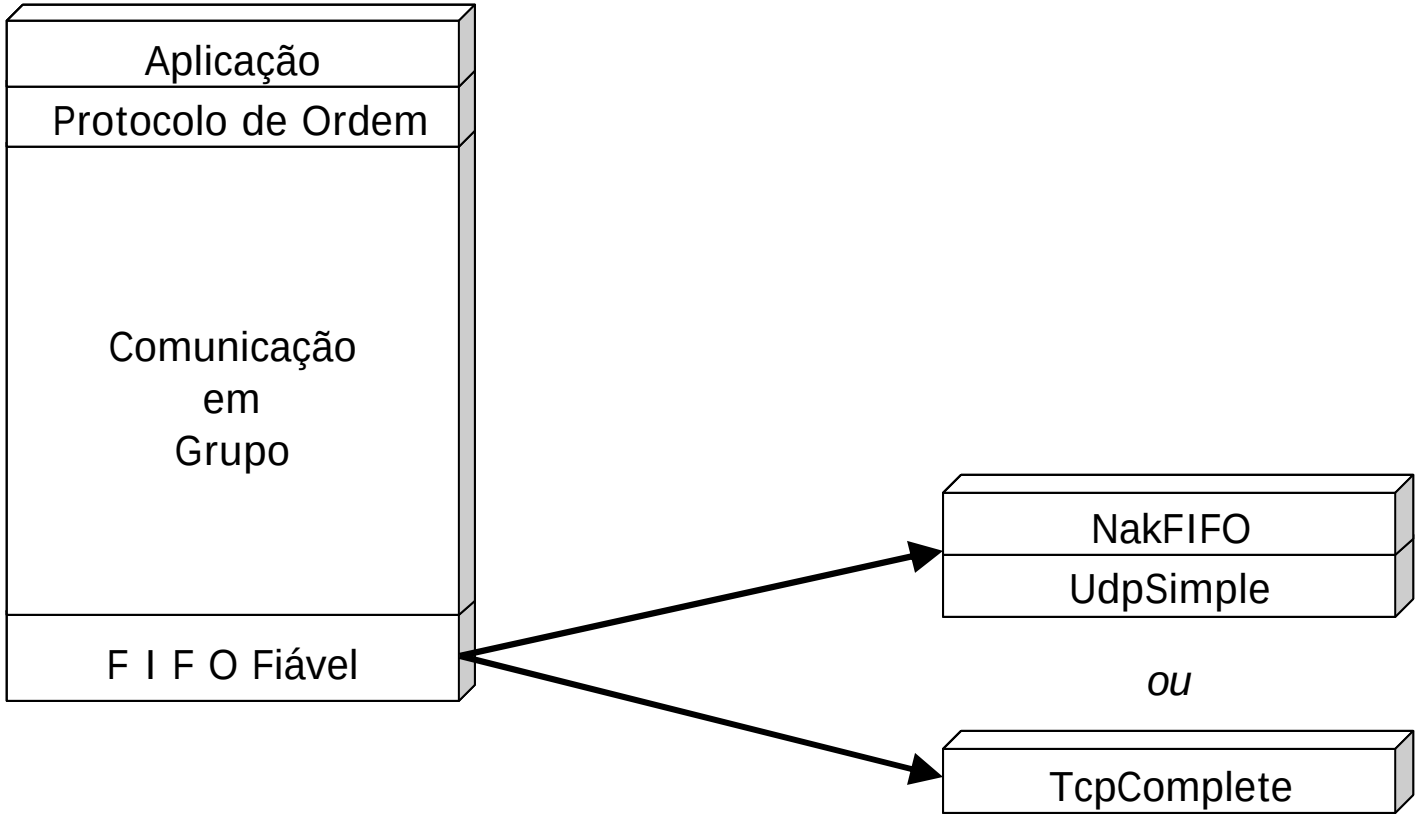
© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

Comunicação em Grupo

- Eventos comuns importantes (package `org.continuent.appia.protocols.group.events`):
  - `GroupInit`
    - Informação para a constituição da vista inicial.
    - Só após o envio deste evento é que o suporte à comunicação em grupo começa a funcionar
  - `GroupEvent`
    - *Superclasse* de todos os eventos utilizados entre as camadas de comunicação em grupo.
  - `GroupSendableEvent`
    - *Superclasse* de todos os eventos com a mensagens a enviar para o grupo.
    - Interface `Send`
      - Um evento a enviar apenas para um sub-conjunto dos membros da vista actual deve concretizar este interface.
      - As ordens na vista dos membros a que se destina devem ser colocadas no campo `dest`.

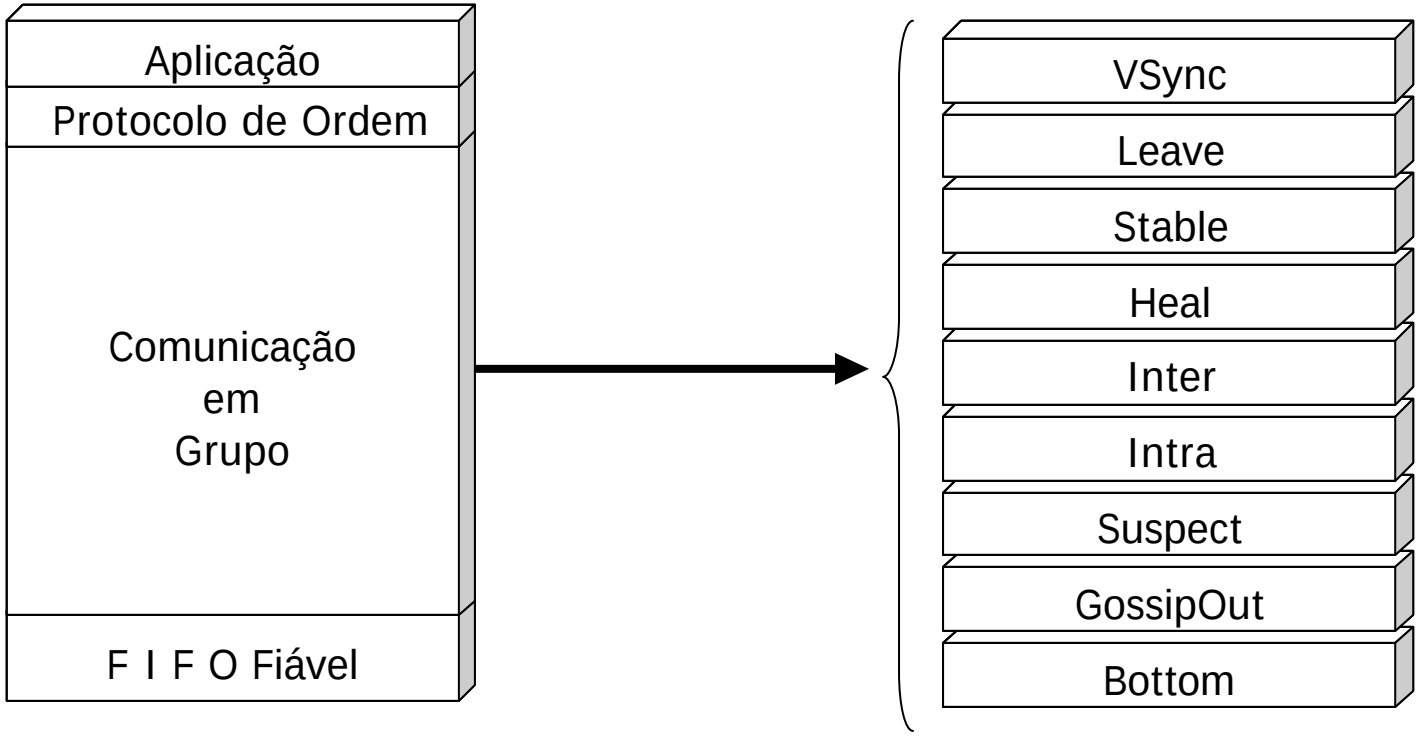
© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

Composição da Com. em Grupo (I)



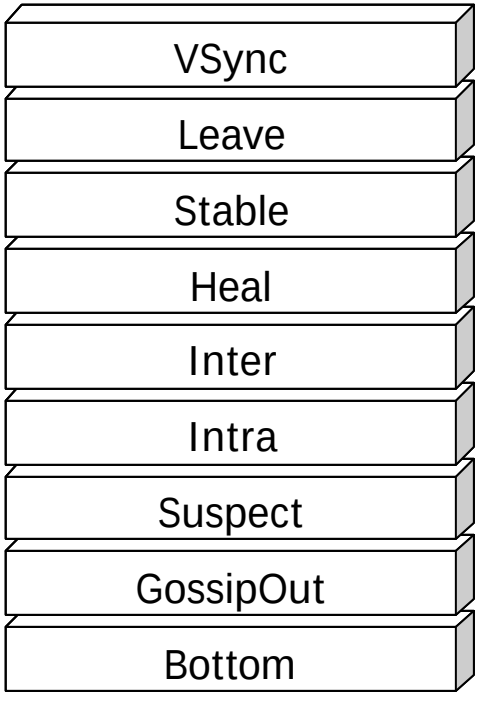
© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

Composição da Com. em Grupo (I)



© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

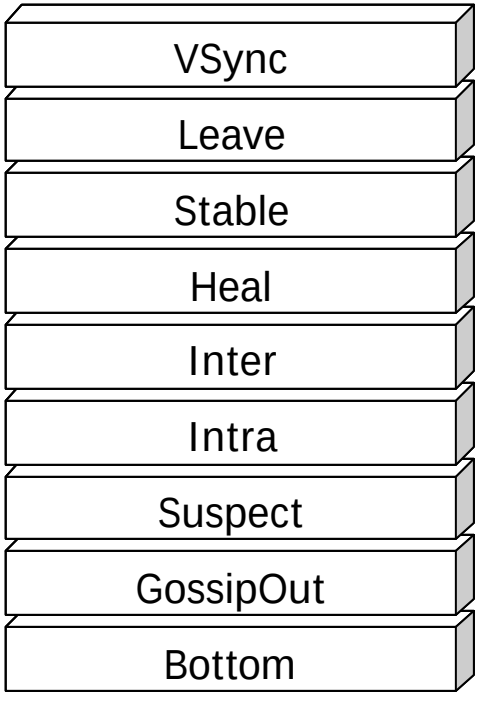
Protocolos da Com. em Grupo (I)



- Bottom → `group.bottom.GroupBottom`
  - Traduz identificações de rede para identificações de grupo → `Endpt` e `Group`
  - Filtra eventos que não pertencem ao grupo ou à vista corrente
    - Anuncia a recepção destes eventos através do evento `OtherViews`.
  - Se necessário, efectua a desmultiplexagem do evento pelos destinatários.

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

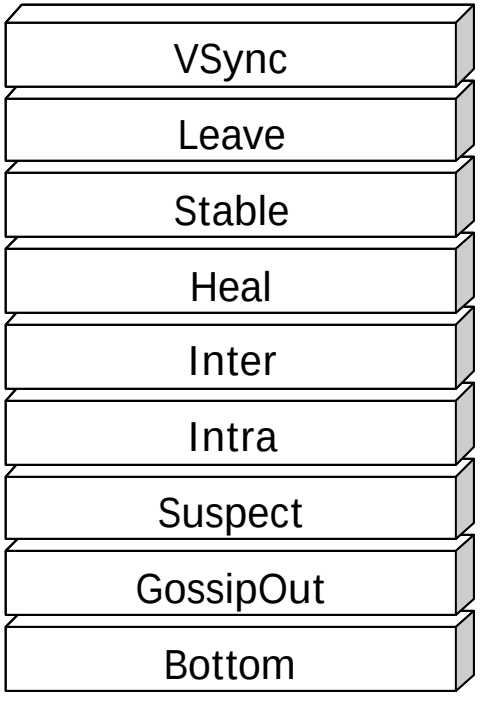
Protocolos da Com. em Grupo (II)



- Suspect → `group.suspect.Suspect`
  - Detector de falha.
  - Usa mecanismos existentes nas camadas de **FIFO Fiável**
  - Protocolo próprio baseado no envio e recepção periódica de mensagens.
    - Reutiliza mensagens enviadas por camadas superiores.
  - Suspeitas são propagadas a todo o grupo.
  - Falha anunciado pelo evento `Fail`.

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

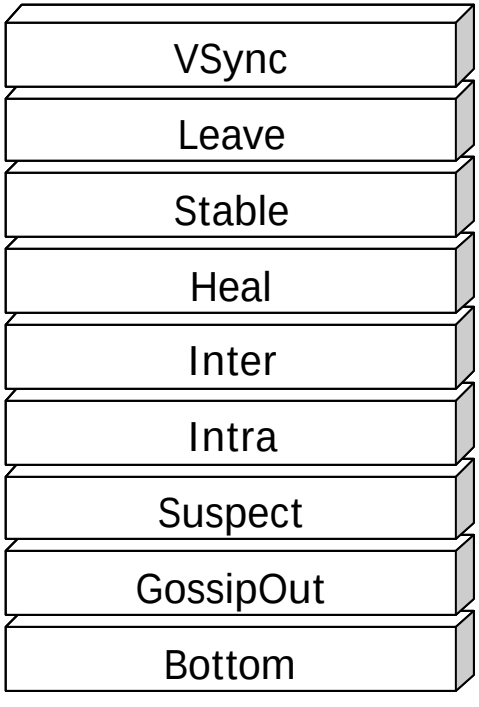
Protocolos da Com. em Grupo (III)



- Intra → `group.intra.Intra`
  - Gere a mudança de vista.
    - Mudança de vista dividida em três passos, executados por outras camadas:
      1. Garantir correcção da mudança
         - Evento `NewView` → VSync
      2. Determinação da composição
         - Evento `PreView` → Inter, Leave
      3. Entrega da nova vista
         - Evento `View` → *todas*
    - Eventos enviados do topo para baixo
  - Mudança iniciada por:
    - Falha de elemento → evento `suspect.Fail`
    - Pedido → evento `intra.ViewChange`
  - Activa apenas no coordenador da vista.

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

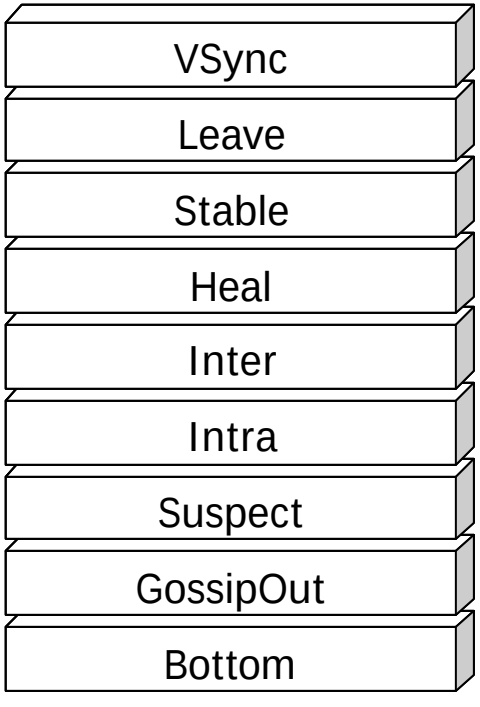
Protocolos da Com. em Grupo (IV)



- Inter → `group.inter.Inter`
  - Une várias vistas concorrentes do mesmo grupo.
    - Detectadas pelo camada Heal
      - `ConcurrentViewEvent`
  - Activa apenas no coordenador da vista.
  - Executa algoritmo de consenso (por inundação) para acordar na composição da nova vista.
    - Participam apenas os coordenadores das vistas.
    - Como novas vistas concorrentes podem ser detectadas, podem ser necessários várias rondas.
    - A falha de um participante implica o reinicio do algoritmo.
  - Para optimizar o número de mudanças de vista, introduz atraso numa mudança de vista em curso para maximizar o número de vistas a unir.

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

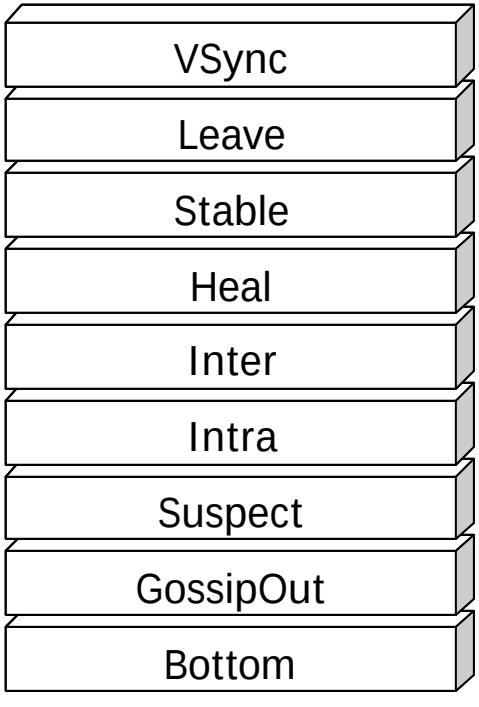
Protocolos da Com. em Grupo (V)



- Heal → `group.heal.Heal`
  - Detecta a existência de vistas concorrentes do mesmo grupo.
    - Pela recepção de mensagens de uma vista diferente.
    - Pela recepção de anúncios através do mecanismo de *gossip*.
  - Activa apenas no coordenador da vista.
- GossipOut → `group.heal.GossipOut`
  - Comunicação com um mecanismo de *gossip*, que pode ser concretizado de duas formas:
    - Um servidor dedicado, que terá que ser replicado para tolerância a falhas.
      - `org.continuent.appia.gossip.GossipServer`
      - Concretiza um *broadcast* entre os processos conhecidos.
    - Um endereço de IP-Multicast próprio.

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

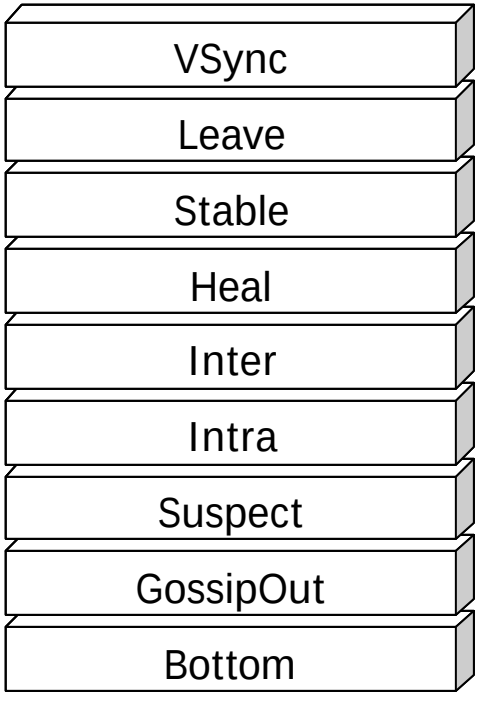
Protocolos da Com. em Grupo (VI)



- Stable → `group.stable.Stable`
  - Retransmite mensagens não recebidas por todos os membros da vista actual.
    - Guarda todas as mensagens recebidas
    - Após detecção da falha de um membro são:
      - Determinadas as mensagens desse membro que não foram recebidas.
      - Um dos membros que recebeu a mensagem em falta é eleito como retransmissor.
      - Enviado pedido ao retransmissor.
  - Para propagação das mensagens recebidas por cada membro, essa informação é enviada de forma periódica num evento `StableGossip`.
  - A informação obtida através do `StableGossip` é também utilizada para limpar as mensagens já recebidas por todos os membros.

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

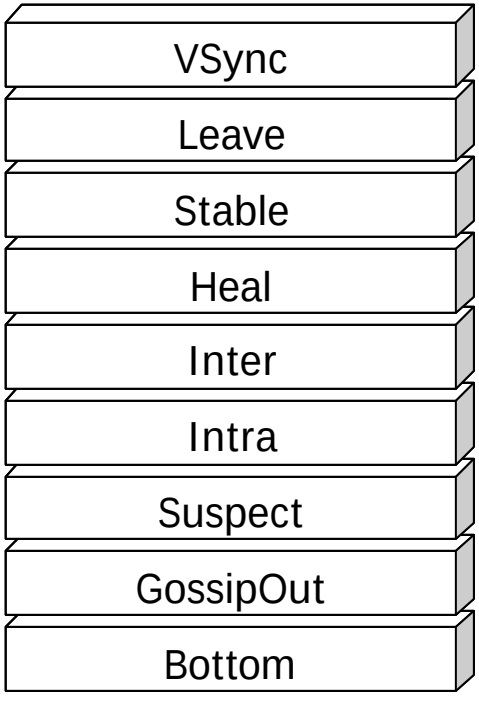
Protocolos da Com. em Grupo (VII)



- Leave → `group.leave.Leave`
  - Efectua a saída graciosa de um membro do grupo.
  - Intenção de sair anunciada através do `LeaveEvent`.
  - Intenção propagada para o coordenador da vista:
    - Mudança de vista é iniciada.
    - Elemento é retirado da composição da nova vista.
    - Elemento recebe `ExitEvent` a informar que o processo de saída está completo.

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia

Protocolos da Com. em Grupo (VIII)



- VSync → `group.sync.VSync`
  - Garante que o processo de mudança de vista respeita a *Sincronia na Vista*.
    - Cada membro conta o nº de mensagens entregues por origem.
    - Quando esses contadores forem iguais em todos os membros activos a mudança de vista pode-se concluir.
  - Para garantir a terminação do algoritmo é necessário que todos os membros do grupo suspendam o envio de mensagens.
    - Suspensão anunciada através do evento `BlockOk`
      - Camadas podem reter o evento para enviar dados *urgentes* após o que devem reenviá-lo.
  - Para optimizar o nº de mensagens trocadas o coordenador da vista centraliza a informação dos vários membros e divulga os valores finais.

© 2006-2009 Alexandre Pinto, DI-FCUL, Reprodução proibida sem autorização prévia