

Especificação de um Protocolo para Negociação de Qualidade de Serviço em Sistemas Multimídia

L.A. Guedes

Departamento de Engenharia Elétrica
Centro Tecnológico
Universidade Federal do Pará
Belém, PA-Brasil

E. Cardozo

Dep. de Eng. de Computação e Automação Industrial
Faculdade de Engenharia Elétrica
Universidade Estadual de Campinas
Campinas, SP-Brasil

e-mail: {affonso, eleri}@dca.fee.unicamp.br

Resumo

A noção de Qualidade de Serviço (QoS) vem recebendo crescente atenção por parte de desenvolvedores e usuários de aplicações multimídia, a ponto de torná-la uma característica chave em tais de sistemas. QoS não é uma concepção nova, sendo usualmente empregada no domínio de redes de computadores para especificar um conjunto de parâmetros tipicamente relacionados às conexões de transporte. Via de regra, QoS é estabelecida através de negociação entre usuários e provedores de serviços. O processo de negociação é relativamente simples se os recursos são gerenciados por uma entidade centralizada (sistema operacional, por exemplo) ou por um conjunto de entidades que empregam protocolos simples de negociação. Infelizmente, em sistemas multimídia distribuídos a negociação e gerência de recursos é uma tarefa não trivial, uma vez que os recursos existentes são bastante diversificados, dispersos e mantidos por diferentes entidades. Para transpor estas dificuldades, é proposto neste artigo um protocolo de negociação de QoS para ambientes distribuídos abertos baseado em sistemas multi-agentes. A sua especificação é realizada utilizando-se as linguagens formais SDL e MSC.

Palavras-chaves: qualidade de serviço, especificação formal, sistemas multimídia, SDL, MSC.

Abstract

Quality of Service (QoS) is receiving strong attention due to its key role in distributed multimedia computing. QoS is not a new concept, being usually employed in the domain of computer networks to specify a set of parameters typically assigned to transport connections. As a rule, QoS is established through negotiation between service users and providers. The process of negotiation is simple if the resources are managed by a centralized entity (e.g. an operating system) or by a set of homogeneous entities such as a network protocol. Unfortunately, in distributed multimedia applications the negotiation and management of resources is a difficult task since resources are very diversified, dispersed and maintained by heterogeneous entities. To cope with diversity and distribution, an agent based approach for QoS negotiation and management in open distributed environments is proposed in this paper. Its specification is defined in the formal languages SDL and MSC.

Keywords: quality of service, formal specification, multimedia systems, SDL, MSC.

1. Introdução

Sistemas multimídia se caracterizam pela capacidade de manipulação de vários tipos de meios, desde os mais simples como textos e gráficos (meios estáticos), até os mais ricos, como animação, áudio e vídeo (meios contínuos). As aplicações multimídia são freqüentemente distribuídas, uma vez que seus componentes se encontram localizados em diferentes nós de processamento de uma rede local ou de longa distância. O uso de meios contínuos em sistemas computacionais distribuídos, diferentemente dos meios estáticos, implica em capturar, processar, transmitir e exibir grandes volumes de dados continuamente e por longos períodos de tempo. Além disso, os dados devem ser capturados e exibidos de forma cadenciada ao longo do tempo, sendo que o intervalo de tempo transcorrido entre a captura e exibição de um dado (atraso) deve ser sempre inferior a um certo valor previamente estabelecido (atraso máximo).

Devido a essas características, os problemas em sistemas multimídia estão relacionados principalmente à gerência de informação demandada pela integração de várias formas de meios em uma infra-estrutura distribuída de

computação e comunicação. Entretanto, há pouca experiência relacionada ao projeto e à implementação de sistemas multimídia distribuídos (SMD). Esta carência se deve basicamente à dificuldade de manipulação dos meios contínuos com as atuais tecnologias de computação e comunicação, somada à grande variedade de padrões existentes para capturar, codificar, transmitir e exibir esses tipos de meios. Assim, SMD requerem implementação de políticas de gerência de recursos orientadas a reserva, monitoramento, otimização e padronização, que em última análise estão relacionadas à qualidade de serviço (QoS) da aplicação. E como SMD executam através de domínios autônomos, a gerência de recursos deve ser baseada sob mecanismos de negociação, sendo a flexibilidade e adaptabilidade características desejáveis.

As primeiras abordagens surgidas para a negociação de QoS possuíam estruturas centralizadas [Cam 93]. Mais recentemente, surgiram propostas baseadas nos paradigmas de mercado [Nah 95] e cliente/servidor [Ker 94]. No esquema de mercado um protocolo de negociação interliga os gerenciadores de recursos localizados em cada nó da rede. Um gerenciador de recursos pode atuar tanto como um comprador, quando recebe recursos, quanto como um vendedor, quando oferece recursos. Já no enfoque de cliente/servidor, a aplicação (cliente) interage com um conjunto de gerenciadores de recursos (servidores), para propósito de negociação e gerência de QoS. Porém, não é claro se estas abordagens são suficientemente flexíveis e dinâmicas ao ponto de atender aos requisitos dos SMD. Em [Aur95] é realizado um excelente trabalho comparativo sobre as propostas mais relevantes de arquitetura para suporte de QoS em SMD.

Nós acreditamos que negociação e gerência de recursos em sistemas multimídia distribuídos devam ser realizadas sob esquemas mais abrangentes, que combinem autonomia, mobilidade e conhecimento. Sistemas Baseados em Agentes (SBA) empregando tanto agentes fixos como móveis [Nwa 96] são capazes de prover tal combinação. Assim, neste trabalho propomos uma arquitetura de negociação e gerência de recursos baseada em SBA, sendo apresentada a especificação do seu protocolo de negociação nas linguagens formais SDL [Bel 91] (Specification and Description Language) e MSC [Rud 96] (Message Sequence Charts).

O restante deste trabalho está dividido da seguinte forma: primeiramente é apresentada uma visão geral sobre QoS em sistemas multimídia, para em seguida serem descritas as características desejadas para um negociador de QoS. Então, é proposto um modelo de negociação de qualidade de serviço baseado em SBA, apresentando a especificação do protocolo de negociação em SDL e MSC. Finalmente, são apresentadas as conclusões sobre o trabalho, além de indicações de desenvolvimentos futuros.

2. Conceitos de Qualidade de Serviço (QoS)

O termo qualidade de serviço (QoS) é bastante intuitivo para o usuário de aplicações multimídia: expressa basicamente quão bons estão os serviços fornecidos pelo sistema. Porém, para avaliarmos o desempenho do sistema é necessário que se disponha de padrões mensuráveis. Assim, a qualidade de serviço de sistemas multimídia é objetivamente descrita a partir de um conjunto de parâmetros bem definidos, os quais buscam caracterizar satisfatoriamente a noção subjetiva de qualidade de serviço fornecida por um dado sistema multimídia. A qualidade de serviço de um determinado fluxo em uma aplicação é claramente função da quantidade de recursos destinada a ele. Recursos incluem CPU, largura de banda de transmissão, capacidade de armazenamento, tipos de periféricos e sistemas de software. Meios contínuos são tratados pelos SMD como uma seqüência de segmentos, cada uma consistindo de uma saída de um dispositivo (câmera, microfone, etc.) amostrada a uma razão constante. Estes segmentos demandam algum processamento antes de serem transmitidos ou armazenados, como codificação, filtragem e compressão, por exemplo. Segmentos são armazenados ou transmitidos em formatos preestabelecidos. Para videoconferência, um segmento de vídeo (quadro) de 200x300 pixels, com 8 bits (256 cores) por pixel, amostrado 20 vezes por segundo e compactado no padrão MPEG é uma configuração comum [Fur 94][Vog 95].

Em geral, a partir do ponto de vista do usuário da aplicação, QoS pode ser descrita em termos dos seguintes parâmetros (parâmetros de QoS do usuário):

- resolução: caracteriza a precisão do processo de digitalização de um segmento de fluxo contínuo e é descrito como uma função de bits por segmento e razão de amostragem;
- distorção: mede o nível de perda de informação de um fluxo contínuo e é função de erros de transmissão e perdas devido às estratégias de compressão, por exemplo;

- nível de sincronização: é uma medida de quão estável está a apresentação dos segmentos de um fluxo de meio contínuo (sincronização intra-meio) e entre fluxos relacionados (sincronização inter-meios);
- interatividade: é uma medida de quanto um fluxo pode ser utilizado para trabalho cooperativos on-line.

Usualmente, a aplicação trabalha com opções finitas para o conjunto de parâmetros de QoS e interage com o usuário para escolhê-las. É comum que se trabalhe com descrições subjetivas como "qualidade de definição de áudio de CD" e "qualidade de definição de vídeo pobre", por exemplo.

A partir do ponto de vista do sistema, o conjunto de parâmetros de QoS é bem estabelecido (parâmetros de QoS do sistema), sendo composto por [Vog 95]:

- Atraso fim-a-fim: é o tempo transcorrido desde a captura ou ativação em uma base de dados de um segmento de fluxo contínuo até a sua apresentação;
- Jitter fim-a-fim: é a variação do atraso fim-a-fim;
- Razão de pacotes perdidos (PER): é a porcentagem de pacotes descartados durante o processo de transmissão;
- Razão de bits errados (BER): é a porcentagem de bits errados devido ao processo de transmissão.

De maneira geral, o atraso fim-a-fim afeta a interatividade da aplicação; o jitter fim-a-fim afeta o nível de sincronização e BER/PER têm impacto no grau de distorção. Limites usuais desses parâmetros em teleconferência são [Fuh 94][Vog 95]: 100-250 ms para o atraso fim-a-fim; 5-10 ms para o jitter fim-a-fim; 0,001-0,01% para o PER e 0,01-0,1% para o BER. Quanto menores forem os valores destes parâmetros maiores serão os índices de qualidade de serviço da aplicação.

Para prover qualidade de serviço, uma aplicação deve implementar uma política eficiente de reserva e gerência de recursos. Neste artigo nós propomos um protocolo para negociação e gerência de QoS que utiliza uma combinação de agentes móveis e fixos. O protocolo não pode garantir deterministicamente que os níveis de QoS negociados serão respeitados ao longo de toda aplicação, porém, acreditamos que o mesmo tenha um desempenho melhor do que as estratégias correntes baseadas em melhor-esforço [Cam 93].

3. Negociação e Gerência de QoS

Por negociação de QoS entende-se definir política de alocação de recursos e seleção de configuração da aplicação. Este processo, devido à natureza conflitante de requisitos, não é trivial. Assim, devido à complexidade da atividade de gerência e negociação de QoS, criou-se uma entidade denominada negociador de QoS, que visa atuar no sistema de modo a melhor atender aos requisitos de qualidade de serviço da aplicação.

Para compreender mais claramente as atribuições de um negociador/gerenciador de QoS é importante conhecer o ciclo de vida de uma sessão multimídia. Um ciclo de vida típico é composto de três fases (estabelecimento, atividade e encerramento) [Haf 95] que são constituídas pelas seguintes etapas:

- Fase de Estabelecimento
 - Especificação de QoS;
 - Mapeamento de QoS;
 - Negociação de QoS;
 - Reserva de Recursos.
- Fase de Atividade
 - Monitoramento de QoS;
 - Adaptação de QoS;
 - Renegociação de QoS.
- Fase de Encerramento
 - Término da sessão.

Com relação à fase de estabelecimento, primeiramente temos a especificação de QoS relacionada aos fluxos de dados da aplicação multimídia. Esta especificação é descrita através dos parâmetros de QoS do nível do usuário, determinando-se os valores desejados e os mínimos tolerados. Depois, esses parâmetros são mapeados em

parâmetros de QoS do nível de sistema e é realizada a negociação da qualidade de serviço requerida pela aplicação, com conseqüente alocação inicial de recursos.

No estado de monitoramento, o sistema mede a qualidade de serviço a partir dos parâmetros de QoS do nível do sistema e atualiza o estado da qualidade de serviço fornecida pelo sistema.

O sistema entra no estado de renegociação de QoS devido a violação de índices de QoS mínimos previamente estabelecidos ou por solicitação de renegociação de QoS por parte do usuário. Aqui, são renegociados os critérios de qualidade de serviço de forma similar ao que ocorre na fase de estabelecimento de sessão (exceto que agora a aplicação continua sua atividade).

No estado de renegociação de QoS os recursos do sistema, como tempo de CPU, largura de banda de transmissão e tamanhos de *buffers*, são redistribuídos entre os diversos fluxos de informação que constituem a aplicação. Esta redistribuição de recursos é realizada a partir de uma política previamente estabelecida.

Já a fase de encerramento de uma sessão multimídia é composta basicamente pela liberação de recursos e sinalização de fim de sessão (término de sessão).

A partir das análises realizadas das fases de uma sessão multimídia, podemos visualizar dois aspectos fundamentais relacionados a esta abordagem para suporte à qualidade de serviço em aplicações multimídia. Estes aspectos são: i) caracterização de qualidade de serviço a partir de parâmetros do sistema e ii) negociação de qualidade de serviço a partir de recursos disponíveis no sistema. A negociação de qualidade de serviço está presente tanto na fase de estabelecimento quanto no decorrer da sessão.

Portanto, após estas considerações sobre negociação de qualidade de serviço, podemos dizer que o negociador de QoS deverá ter a capacidade de balancear a carga entre os componentes da aplicação, encontrar, reservar e liberar recursos, além de monitorar a qualidade do serviço fornecida pelo sistema. Para tanto, é desejado que o negociador de QoS tenha estrutura descentralizada e liberdade de negociação, além de permitir a reconfiguração da aplicação e até mesmo alterar os objetivos da negociação de QoS, caso necessário.

4. Negociador de QoS Baseado em SBA

Sistemas baseados em agentes (SBA), ou multi-agentes, são sistemas de computação, onde as entidades básicas de computação são agentes [Nwa 96] que trabalham cooperativamente na solução de problemas, muitas vezes em ambientes distribuídos. Por sua vez, agentes são entidades autônomas que possuem estado, estrutura e objetivos próprios. O estado corresponde aos dados associados ao agente. A estrutura é a programação do agente, que corresponde às habilidades do agente. E por fim, os objetivos dizem respeito às atividades que o agente pode ou deve desempenhar para resolver uma determinada tarefa. O local onde os agentes habitam e trocam informações é denominado de agência.

Inicialmente, SBA foram utilizados em resolução distribuída de problemas na área de Inteligência Artificial. Mais recentemente, SBA vêm despertando grande interesse de aplicação na área de computação móvel. Nessa área, a principal característica requerida de um agente é a mobilidade [Che 95], ou seja, a capacidade de executar parte de seu código em uma agência e, se necessário, migrar para outra, onde continua a execução do ponto subsequente à migração.

A arquitetura proposta aqui para negociação e gerência de QoS é baseada em sistemas multi-agentes e é composta de uma agência de QoS para cada nó (estação) da aplicação, a qual suporta dez tipos de agentes, sendo três com capacidade de migração e sete fixos. Além dos agentes, a agência possui contratos e uma fábrica de agentes. Uma ilustração de uma agência de QoS, com seus componentes e suas interligações é mostrada em Fig. 1.

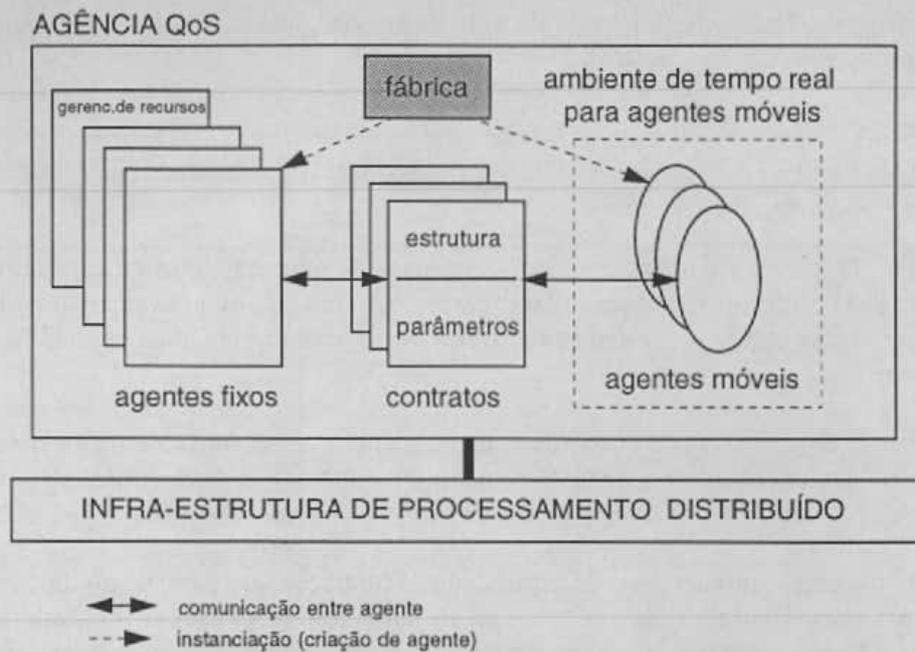


Fig. 1- Arquitetura baseada em SBA para gerência de QoS.

O Contrato é composto de estrutura e parâmetros, onde a estrutura diz quais fluxos compõem a aplicação, enquanto os parâmetros referem-se às características dos fluxos (níveis aceitáveis de qualidade de áudio, por exemplo). Assim, quem propõe um contrato deve especificar o que deseja receber e o que pode fornecer, ou seja, quais recursos de hardware e software possui e deseja. Para tal, o protocolo de negociação de contrato deve suportar as seguintes atividades: proposta, contraproposta e aceitação ou não de contrato.

A fábrica de agentes é um componente da agência responsável pela instanciação (criação) de agentes.

A agência suporta as seguintes classes de agentes: agente de interface, agente mapeador_de_qos, agente negociador_local, agente estimador_de_recursos, agente gerente_de_recursos, agente negociador_de_contrato, agente renegociador_de_contrato, agente monitor_de_contrato, agente monitor_de_qos e agente adaptador_de_qos. Alguns agentes são fixos, isto é, executam integralmente na agência em que foram instanciados. Outros agentes são móveis, isto é, executam parte de seu código numa agência e migram para outra, onde reiniciam sua execução ou continuam sua execução do ponto posterior à migração, de modo a cumprirem seus objetivos.

O agente de interface é fixo e interage com o usuário da aplicação para fins de especificação de qualidade de serviço em contratos na fase de estabelecimento de sessão e para informar os níveis de QoS durante a fase de atividade da aplicação.

O mapeador_de_qos não possui capacidade de migração e tem as seguintes atribuições: i) na fase de estabelecimento de sessão, ele mapeia especificações de QoS do usuário em especificações ao nível de sistema e ii) na fase de atividade (execução) ele interage com o agente monitor_de_qos realizando o mapeamento inverso, de modo a informar ao agente de interface sobre o desempenho da aplicação.

O negociador_local não possui capacidade de migração. Ele é o responsável pela negociação e renegociação local dos contratos em cada agência envolvida na aplicação, de tal modo que ele permanece ativo por todo o ciclo de vida de uma sessão.

O negociador_de_contrato é um agente com capacidade de migração. Quando criado, persiste durante toda a fase de estabelecimento de sessão. Antes de ser extinto, no final da fase de estabelecimento de sessão, este agente gera uma cópia de contrato em cada agência participante da sessão. Sua função é negociar o contrato com todos os participantes da aplicação, valendo-se de um protocolo de negociação bem definido. Ele interage com os agentes negociador_local de cada agência envolvida na aplicação.

O agente estimador_de_recursos não tem capacidade de migração. Ele recebe requisições do negociador_local e realiza estimativas de recursos locais de modo que atenda às requisições de QoS solicitadas.

Este agente permanece ativo durante todo o ciclo de vida da sessão, pois ele pode receber requisições tanto para propósito de negociação quanto de renegociação

O `gerente_de_recursos`, também fixo, permanece ativo durante todo o ciclo de vida da sessão e tem por função apenas alocar e liberar recursos a partir de solicitações do `estimador_de_recursos`, `negociador_local` e `adaptador_de_qos` (na fase de atividade da sessão).

O `renegociador_de_contrato` é um agente com capacidade de migração, que atua apenas na fase de execução da sessão. Ele é criado pelo usuário, caso deseje alterar parâmetros de QoS, ou pelo `adaptador_de_qos`, caso alguma violação severa de parâmetros de QoS seja detectada. Antes de ser extinto, ele atualiza todas as cópias de contrato nas respectivas agências.

O agente `monitor_de_qos` não tem capacidade de migração e a sua função é medir o desempenho local do sistema a partir de medidas dos parâmetros de QoS do nível de sistema. Ele é criado no início da fase de atividade da sessão.

O `monitor_de_contrato` é um agente com capacidade de migração que é criado no início da fase de atividade da sessão e há um para cada contrato. Ele tem por função monitorar se os fluxos multimídia nos destinos estão cumprindo os contratos. Para tal, ele usa o seguinte protocolo: i) um agente `monitor_de_contrato` ao chegar em uma estação *i* informa ao agente `monitor_de_qos` como estão sendo exibidos os fluxos dessa estação nas outras estações participantes da aplicação, ii) consulta o agente `monitor_de_qos` local para saber como estão chegando os fluxos das outras estações nessa estação e iii) parte para outra estação.

Cada estação envolvida na aplicação ainda possui um agente `adaptador_de_qos` fixo, que é criado no início da fase de atividade da sessão e tem atribuição de interagir com os agentes `monitor_de_qos` e `gerente_de_recursos` da seguinte forma: dependendo da informação que obtém do `monitor_de_qos`, ele pode apenas efetuar uma ação de adaptação de QoS localmente (interagindo com o `gerente_de_recursos` local) ou pode criar um agente de renegociação de contrato com o intuito de buscar uma renegociação global.

Uma renegociação de contrato é proposta para uma das seguintes finalidades: mudança de parâmetros e/ou mudança de estrutura de contrato. Uma renegociação de contrato para mudança de parâmetros pode solicitar aumento de QoS, possivelmente sugerida pelo usuário, ou uma diminuição de QoS, usualmente promovida pelo sistema em decorrência de sobrecarga. Já uma renegociação de contrato para mudança de estrutura se destina a inclusão/exclusão de fluxos na/da aplicação, geralmente acarretando uma renegociação de parâmetros do contrato.

Para que não haja conflito na negociação, todos os agentes móveis viajam utilizando sempre um mesmo sentido de rota, para efetuar esta ordenação na rota de viagem pode-se utilizar os números IP das máquinas envolvidas na aplicação, por exemplo.

Além de uma agência de QoS em cada nó da aplicação, o modelo ainda prevê um Trader no sentido ODP [ODP 95], com o propósito de armazenar informações sobre os recursos e serviços das agências de QoS. Estas informações seriam de caráter estrutural, como tipo de dispositivos de entrada e saída, tipo dos serviços de transporte, infra-estrutura de rede e serviços de compressão de áudio e vídeo, por exemplo. No caso, somente participarão de aplicações as agências que estiverem cadastradas no Trader.

Na próxima seção será apresentada a especificação em SDL e MSC do protocolo de negociação de QoS do nosso modelo baseado em SBA.

5. Especificação do Protocolo de Negociação de QoS

Nesta seção será apresentada a especificação do protocolo de negociação de contrato, descrita nas linguagens formais SDL e MSC, ambas padronizadas pela ITU [ITU 96]. A especificação foi implementada na ferramenta GEODE/VERILOG [Geo 95], que suporta essas duas linguagens formais. Por questão de escopo, o protocolo correspondente à parte de gerência de QoS não será tratado neste trabalho.

Durante a fase de negociação de contrato, que corresponde à fase de estabelecimento de sessão, os agentes que estarão ativos nas agências de QoS serão os seguintes: `interface`, `mapeador_de_qos`, `negociador_local`,

negociador_de_contrato, gerente_de_recursos e estimador_de_recursos. Assim, o único agente móvel nesta fase é o negociador_de_contrato.

Na implementação assumimos que cada nó da aplicação possui uma agência de QoS, que corresponde a um bloco em SDL e cada agente, por sua vez, será implementado como um processo SDL. A fábrica também é implementada como um processo SDL.

As visões hierárquica e de interconexão, em SDL, da agência de QoS são mostrada em Fig. 2 e Fig. 3, respectivamente. Da visão hierárquica, Fig. 2, podemos verificar que o sistema (Agencia_QoS) é composto de uma área de declaração de variáveis globais (Pr declaration) e de um único bloco (Agencia_de_QoS), este por sua vez contém os processos SDL. Os processos em SDL são máquinas de estados finitos que trocam mensagens entre si através de rotas de sinais. A troca de mensagem com o ambiente externo ao bloco é realizada através de canais que são conectados às correspondentes rotas. A visualização de com quem e quais sinais cada processo pode trocar é fornecida pela visão de interconexão (Fig. 3), porém esta representação não fornece as possíveis ordens temporal das trocas de mensagens. Os pares ordenados associados aos processos em Fig.2 correspondem respectivamente aos números inicial e máximo de instâncias do processo no modelo.

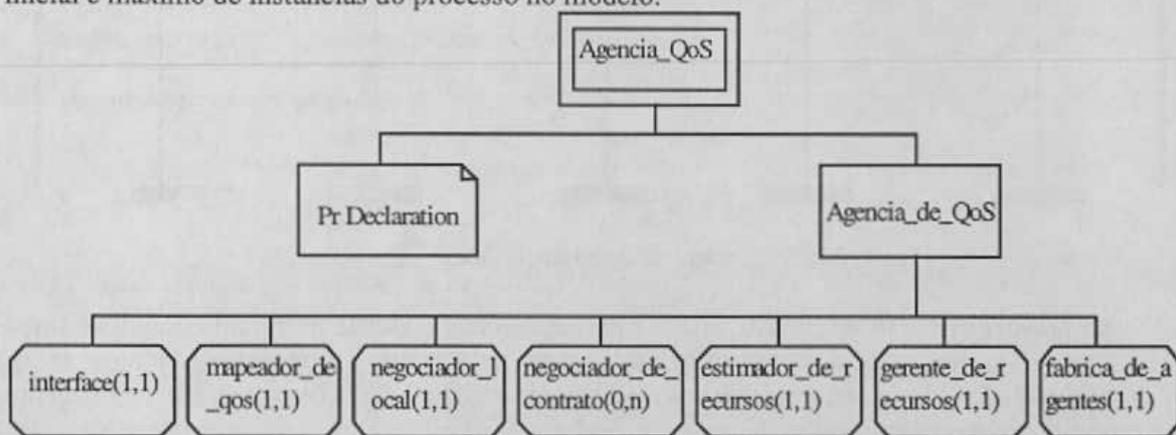


Fig. 2- Visão hierárquica em SDL do modelo da agência de QoS.

Como a visão de interconexão em SDL não é clara sobre as possíveis seqüências temporais de troca de mensagem entre os processos SDL, tornou-se habitual a utilização da linguagem formal MSC (Message Sequence Charts). A MSC é fundamentalmente uma linguagem para descrição de troca de mensagem entre entidades (no caso processos em SDL) e seu ambiente. A principal vantagem da MSC se deve a uma representação gráfica simples e fácil que fornece a compreensão intuitiva imediata do comportamento temporal do sistema. A MSC é utilizada mais especificamente para: i) geração de possíveis cenários de seqüências temporais das trocas de mensagens entre processos e ambiente e ii) simulação e cheque de consistência da especificação em SDL. O primeiro é referente à especificação de requisitos do processo de comunicação do modelo, enquanto o segundo diz respeito à etapa de validação e teste do sistema.

Por questão de simplificação iremos apresentar aqui somente os principais cenários de troca de mensagem especificados para o processo de negociação de QoS, que são divididos em dois grandes grupo: situação de sucesso na negociação e situação de falha na negociação.

5.1. Cenários de Sucesso na Negociação

Um processo de negociação de QoS em uma aplicação multimídia distribuída é composto de três etapas: i) negociação da proposta localmente na agência proponente, ii) verificação de disponibilidade estrutural global, via consulta ao Trader e iii) negociação da proposta com cada agência de QoS envolvida na aplicação. Para uma negociação obter sucesso é necessário que todas as três etapas do processo sejam bem sucedidas. Na primeira etapa há ainda a possibilidade de contraproposta do sistema à proposta inicial, isto ocorre nos casos em que o sistema detecta a impossibilidade de atender localmente aos requisitos da proposta de contrato.

Fig. 3- Visão de interconexão em SDL do bloco Agencia_de_QoS.

Assim, a etapa de negociação de QoS na própria agência proponente do contrato pode ser bem sucedida com ou sem contraproposta do sistema. O primeiro caso é mostrado em Fig. 4, através de MSC, onde as barras verticais mais espessas correspondem ao ambiente.

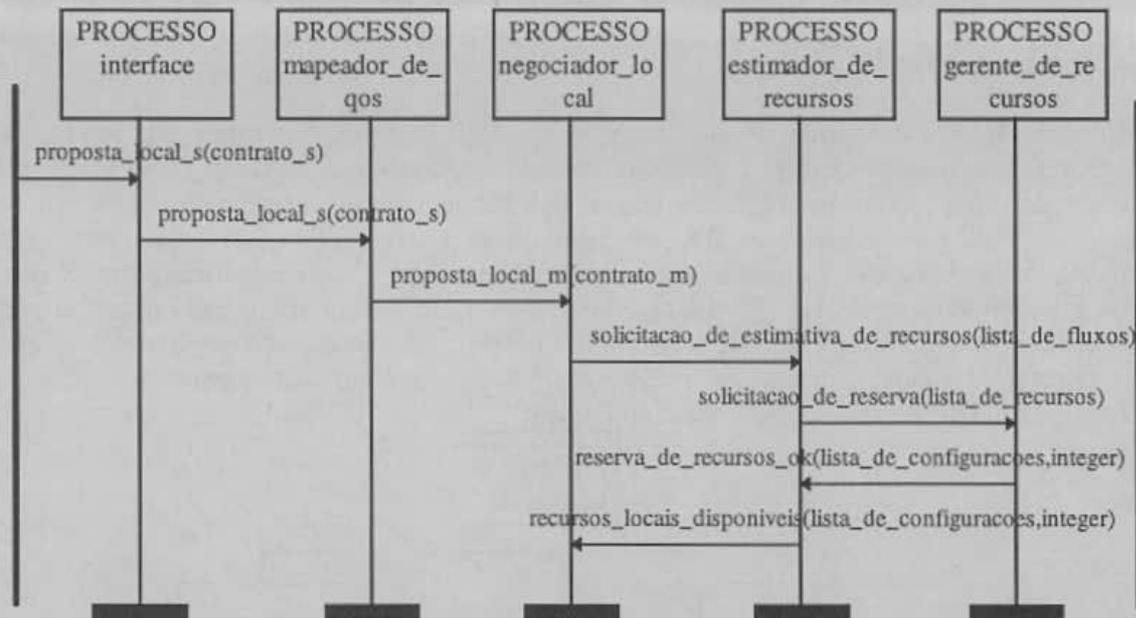


Fig. 4- MSC da etapa de negociação local bem sucedida.

Fig. 4 descreve a situação em que o agente interface recebe do usuário da aplicação, via ambiente, uma proposta de contrato (*proposta_local_s*), descrita em termos de estrutura, fluxos constituintes da aplicação e parâmetros de QoS do nível do usuário descritos em termos de valores subjetivos (*contrato_s*). O agente interface repassa esta proposta ao agente mapeador_de_qos, que por sua vez a retransmite ao agente negociador_lo cal, porém em termos de parâmetros de QoS do nível de sistema descritos de termos de faixas de valores aceitáveis (*contrato_m*). Daí, o negociador_lo cal solicita ao estimador_de_recursos uma estimativa da quantidade de recursos que será necessária para atender aos requisitos de qualidade requeridos pelos fluxos associados com esta agência de QoS (*lista_de_fluxos*). O estimador_de_recursos após efetuar a estimativa solicitada verifica junto ao gerente_de_recursos se há recursos disponíveis. Como a solicitação é viável, o gerente_de_recursos faz a reserva requerida, retornando ao estimador_de_recursos uma lista de possíveis configurações para a aplicação e um inteiro que corresponde ao número da reserva.

Caso seja detectada a inviabilidade da proposta, o estimador_de_recursos propõe uma contraproposta, que pode ser aceita ou não pelo proponente do contrato. Fig. 5 mostra o cenário em que a contraproposta é aceita. No caso, ao ser detectado a inviabilidade da proposta o gerente_de_recursos fornece na contraproposta o máximo de recursos disponíveis com as correspondentes configurações e o número da reserva, daí o estimador_de_recursos produz uma contraproposta de contrato estendido (*contrato_e*), que contém estrutura, parâmetros de QoS do nível de sistema descritos em faixas de valores, além das possíveis configurações para a aplicação. De tal modo que uma contraproposta descrita em termos parâmetros de alto nível (*contrato_s*) chegue ao usuário, que no caso sinaliza com um aceite.

A etapa de verificação de viabilidade de infra-estrutura global do contrato, via consulta ao Trader, para o caso de sucesso é mostrada em Fig. 6. Esta etapa é importante porque garante que somente será disparado um processo de negociação remota caso haja disponibilidade de infra-estrutura para suportar a aplicação.

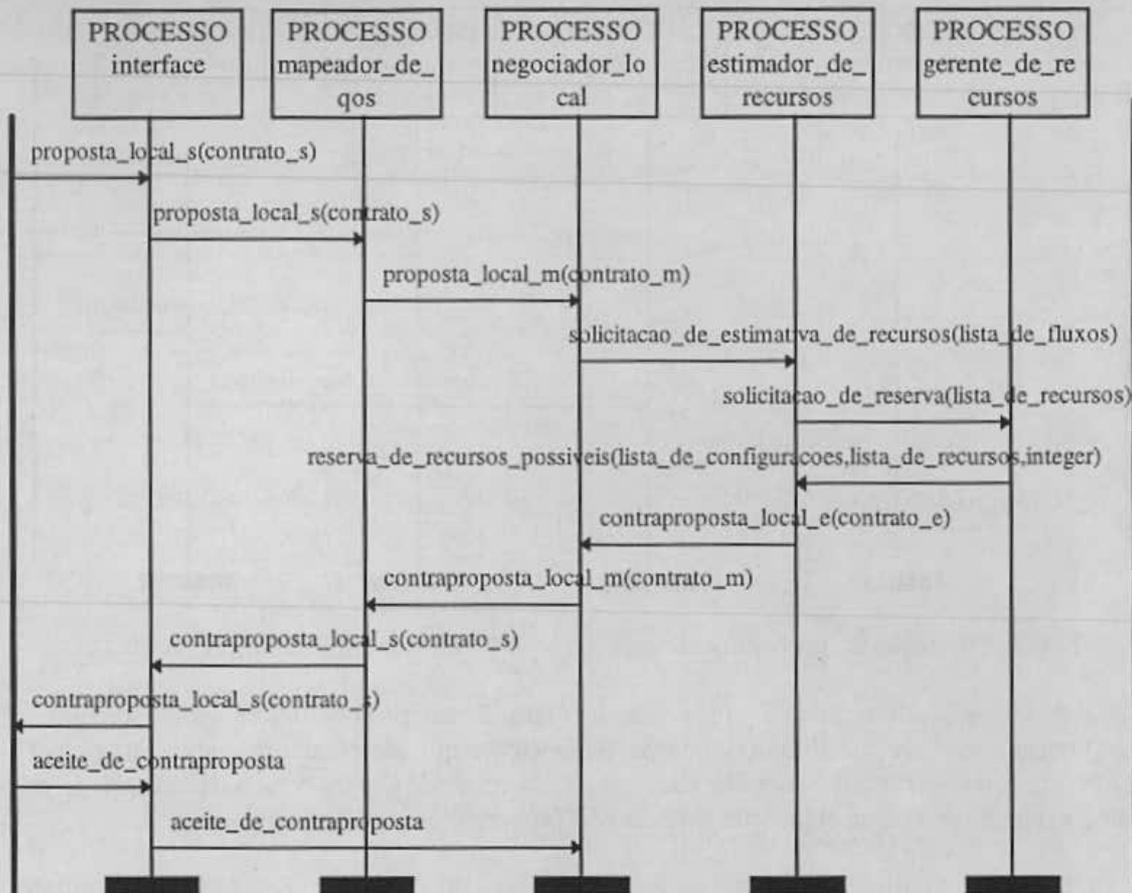


Fig. 5- MSC com aceite de contraproposta de contrato.

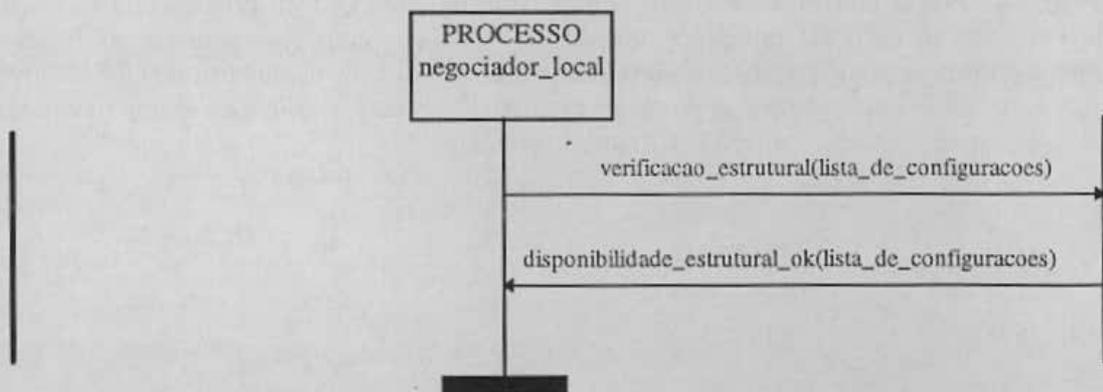


Fig. 6- MSC da verificação de viabilidade de infra-estrutura global da proposta de contrato.

Após esta etapa, a agência proponente do contrato cria um agente negociador_de_contrato móvel, que irá negociar a proposta de contrato em todas as outras agências participantes da aplicação. Fig. 7 mostra o MSC do processo de instanciação do agente negociador_de_contrato. No caso, para efeito de especificação estamos utilizando um mecanismo de replicação que simula o processo de migração do agente negociador_de_contrato, causando o mesmo comportamento lógico no modelo. Como processos em SDL somente podem instanciar outros que estejam no mesmo bloco, a fábrica (fabrica_de_agentes) envia um sinal para uma outra fábrica remota (migracao_de_negociador) solicitando a instanciação de um processo negociador_de_contrato, de tal modo a emular o procedimento de migração desse agente.

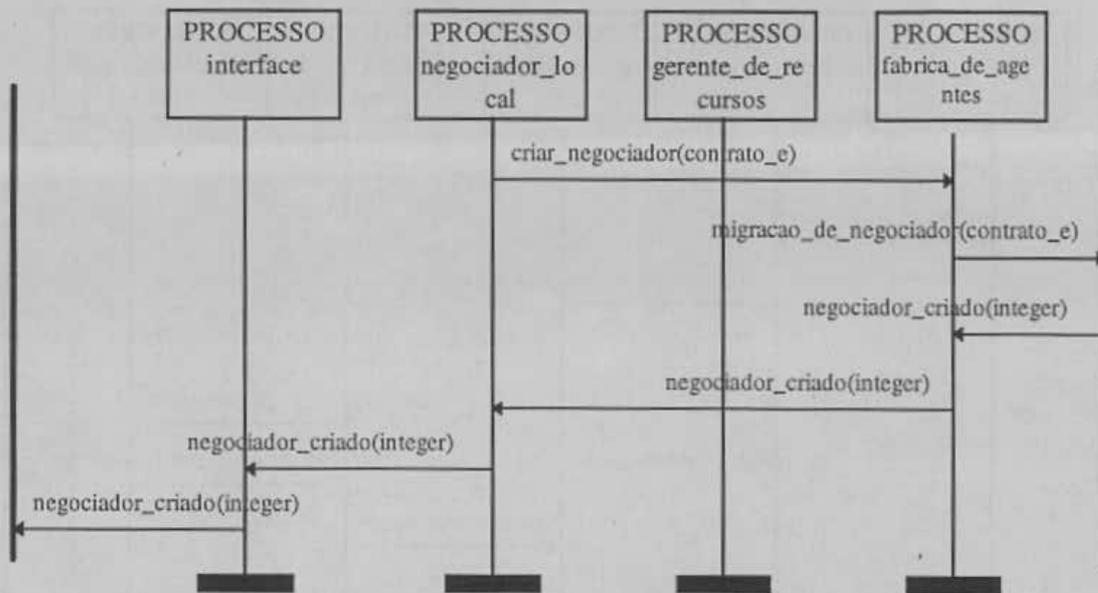


Fig. 7- Emulação do procedimento de migração do agente negociador_de_contrato.

A etapa de negociação remota de proposta de contrato do protocolo terá êxito somente se o agente negociador_de_contrato for bem sucedido em todas as negociações que ele efetuar durante a sua peregrinação pelas agências de QoS que estão envolvidas na aplicação. Fig. 8 mostra o MSC para o caso de sucesso de negociação de contrato em uma agência remota, ou seja, uma agência não proponente de contrato.

Em Fig. 8 a seta pontilhada e o "x" na barra temporal do processo negociador_de_contrato significam respectivamente instanciação e morte do processo. Já o segmento pontilhado na barra temporal de negociador_local corresponde a uma correção, ou seja, região onde a ordem temporal dos sinais é irrelevante.

Observando-se Fig. 8 podemos verificar o mecanismo de emulação do processo de migração do agente negociador_de_contrato, no caso via um procedimento de replicação com passagem de informação através da variável contrato_e (contrato estendido). Esta variável contém a estrutura, os parâmetros de QoS do nível de sistema descritos em faixas de valores aceitáveis e as possíveis configurações para a aplicação. É importante ressaltar que a rota de viagem do agente é extraída a partir da estrutura do contrato.

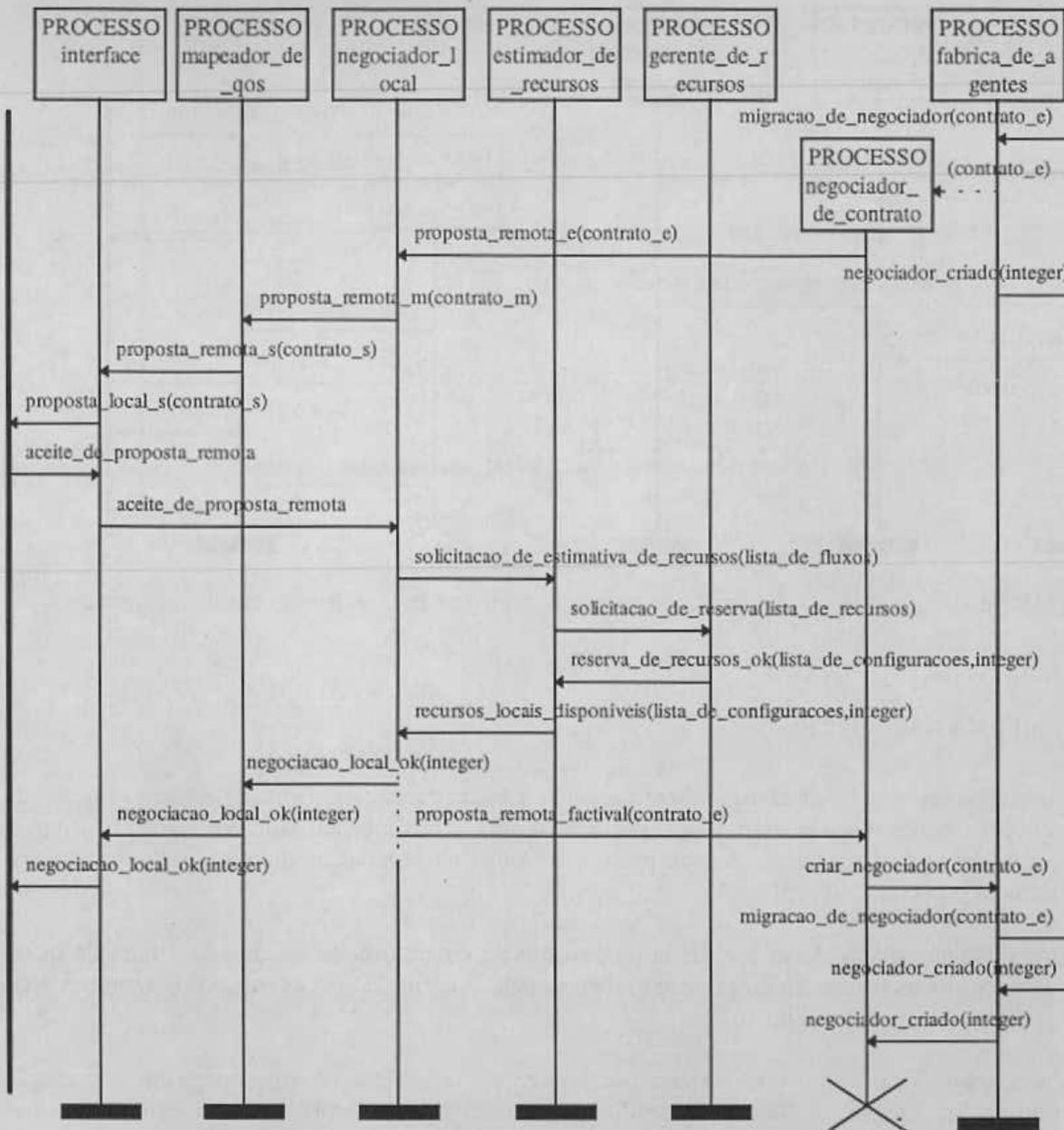


Fig. 8- MSC de uma etapa de negociação de contrato bem sucedida em uma agência remota.

Após o agente negociador_de_contrato ter obtido sucesso em todas as suas negociações, ele volta a visitar todas as agências de QoS remotas, para confirmar a reversa de recursos e habilitá-las para o início da fase de execução de sessão, além de fornecer uma cópia do contrato negociado. Este procedimento é mostrado em Fig. 9. Após este procedimento, o agente negociador_de_contrato volta a agência proponente para efetuar procedimento similar de confirmação de reserva, porém, nessa agência ao final do procedimento de reserva, ao invés de migrar ele irá ser extinto. Caracterizando-se assim o final da fase de negociação de contrato.

Desse modo, todas as agências que irão participar da aplicação se encontram prontas para o início da fase de execução, que irá ser disparada por uma sinalização da agência proponente do contrato.

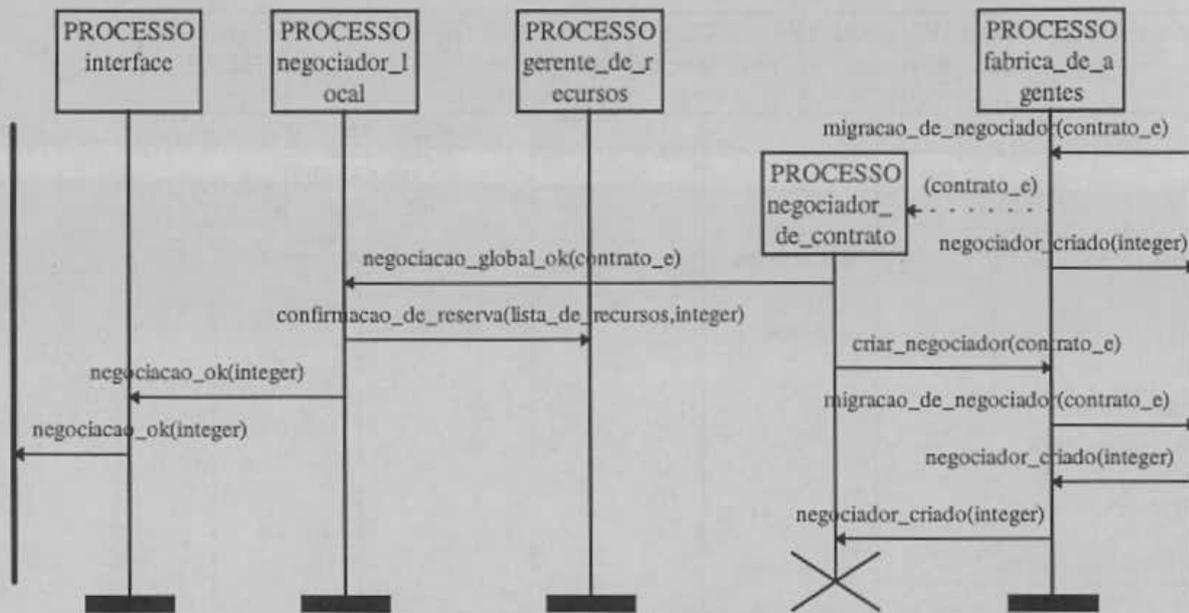


Fig. 9- MSC do processo de confirmação de reserva de recursos da fase de negociação de contrato.

5.2. Cenários de Falha na Negociação

Ao contrário de uma negociação de contrato bem sucedida, para se caracterizar uma negociação fracassada basta que haja falha em pelo menos uma das etapas de negociação descritas na seção anterior. No caso, a ocorrência de falha aborta todo o processo de negociação. Assim, pode-se ter falha na negociação devido a falta de recursos locais ou remotos à agência proponente da aplicação.

A falha na etapa de negociação local à agência proponente do contrato pode ser devido à falta de infraestrutura local para atender aos requisitos da proposta ou à recusa pelo usuário de uma eventual contraproposta do sistema. O MSC do primeiro é exibido em Fig. 10.

Caso obtenha sucesso localmente, que corresponde a Fig. 4 e Fig. 5, a próxima etapa de negociação corresponde à consulta ao Trader, para propósito de verificação de viabilidade da proposta em termos de infraestrutura global. Caso o Trade constate que não há opções que atendam aos requisitos da proposta, a negociação falha. O MSC deste caso é mostrado em Fig. 11. Nessa circunstância, a agência proponente tem de liberar os recursos reservados previamente.

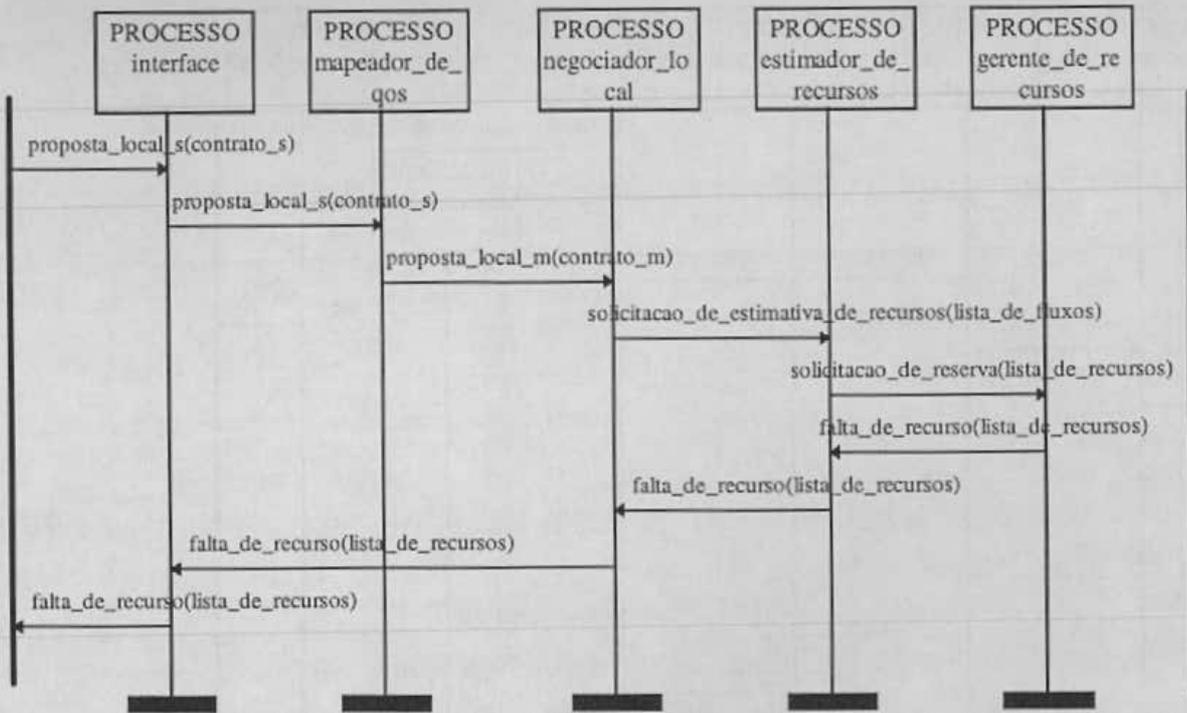


Fig. 10- MSC de um cenário de falha local de negociação devido à falta de infra-estrutura.

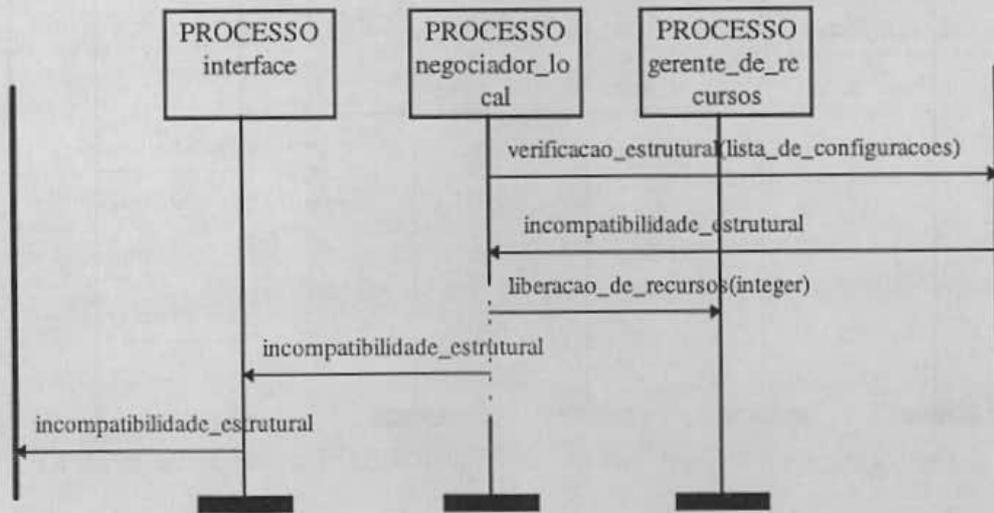


Fig. 11- MSC do cenário de falha de negociação na etapa de consulta ao Trader.

Na última etapa do processo de negociação, ou seja, negociação com todas as agências não proponentes do contrato, há a possibilidade de falha por recusa de alguma agência em participar da sessão ou ainda por falta de recursos para atender aos requisitos da proposta de contrato. O segundo caso é retratado em Fig. 12, que é o cenário de falha correspondente ao de sucesso descrito em Fig. 8. Porém, neste caso (Fig. 12), quando o gerente_de_recurso s detecta que não há disponibilidade de recursos, os sinais gerados provocam a migração do negociador_de_contrato com o objetivo de cancelar eventuais reservas efetuadas em outras agências.

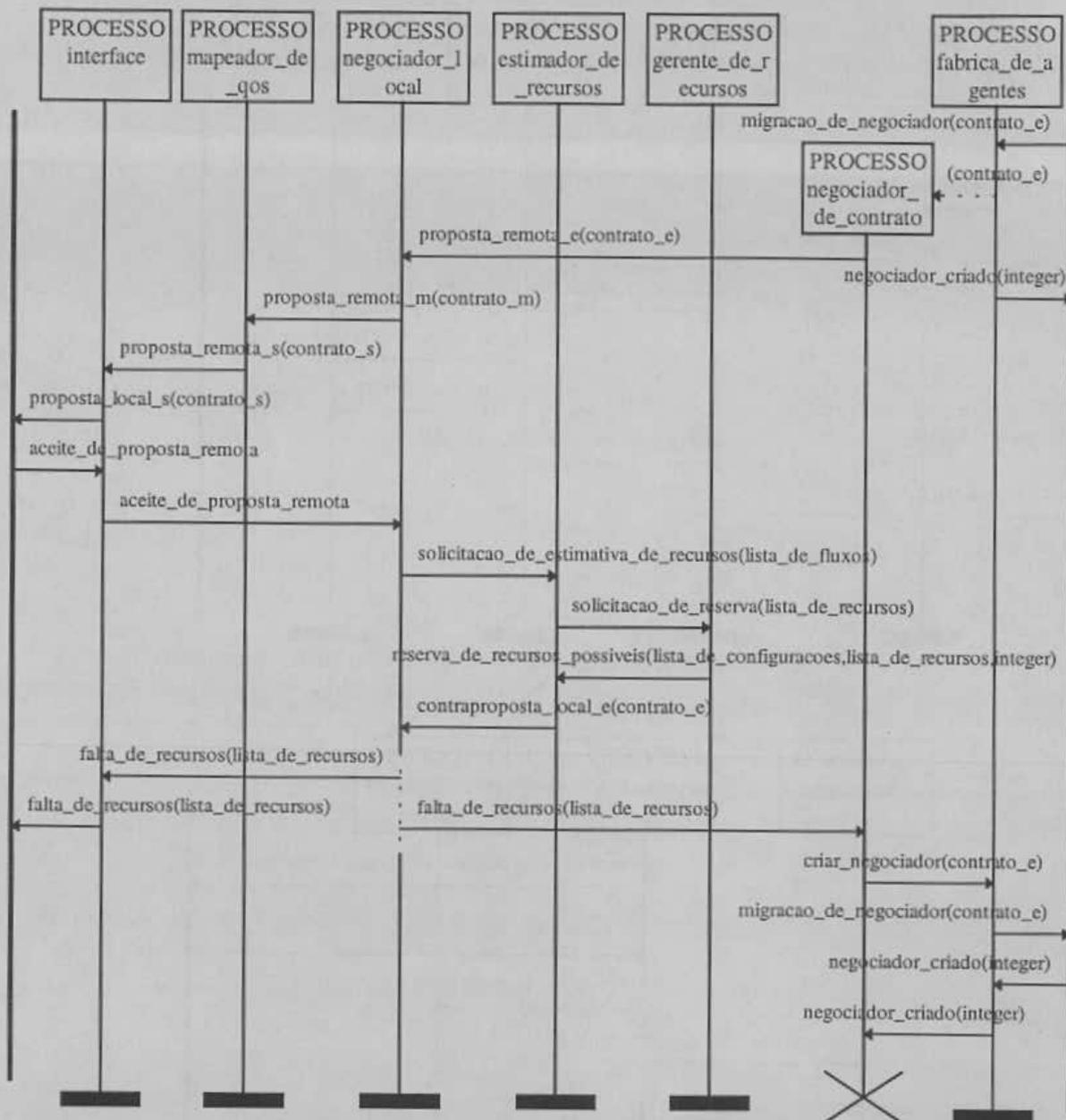


Fig. 12- Cenário de falha de negociação de contrato em uma agência remota.

6. Futuros Trabalhos

Este trabalho corresponde à primeira fase de um projeto que visa a construção e validação de uma arquitetura para suporte de qualidade de serviço em aplicações multimídia distribuída, como videoconferência e vídeo sob demanda. As próximas fases desse projeto são as seguintes:

- Especificação da fase de execução de sessão, utilizando-se também das linguagens SDL e MSC, suportadas pela ferramenta Geode/Verilog.
- Validação do protocolo de negociação de QoS conjuntamente com o de gerência de QoS através de simulação exaustiva, via a ferramenta Geode/Verilog.
- Geração automática de código em C++ do modelo validado, via a ferramenta Geode/Verilog.
- Comunicação distribuída com replicação. Nesta fase iremos interligar instâncias de agências de QoS via o protocolo TCP. Porém, ainda utilizando o mecanismo de réplica para emular o processo de migração de agentes.
- Comunicação distribuída via ORB com replicação. Nesta fase toda comunicação entre elementos do modelo será via um ORB (Object Request Broker), que no caso será a implementação CORBA Orbix[ION 95]. Porém, ainda com o mecanismo de emulação de migração.

- Comunicação distribuída via ORB com mobilidade. Nesta última fase do projeto é prevista a incorporação de suporte a agentes móveis na plataforma CORBA Orbix, de modo que se tenha efetivamente mecanismo de migração de agentes.

A partir daí poderemos utilizar esta arquitetura para desenvolvimento de aplicações multimídia distribuídas e também para validação de desempenho de algoritmos de estimação de recursos, adaptação de QoS e controle de QoS, entre outros.

7. Conclusões

Neste trabalho foi proposta uma arquitetura para negociação e gerência de qualidade de serviço (QoS) para aplicações multimídia distribuídas. Além disto, também foi apresentada a especificação do seu protocolo de negociação de QoS, descrito nas linguagens formais SDL e MSC.

Acreditamos que essa arquitetura, baseada em sistemas de agentes fixos e móveis, seja mais hábil para efetuar o processo de negociação e gerência da qualidade de serviço em aplicações multimídia distribuída, quando comparadas com as baseadas nos paradigmas de mercado e de cliente-servidor.

Outro aspecto fundamental neste trabalho foi a utilização de linguagens formais, via uma ferramenta de software (Geode/Verilog), para a especificação do protocolo de negociação de contrato. Isto permitiu o desenvolvimento seguro e claro da especificação do protocolo, além de permitir que se faça teste de validação nas outras etapas do projeto.

Agradecimentos

Esta pesquisa é financiada em parte pelas seguintes agências brasileiras de fomento à pesquisa: FAPESP (processo 92/3507-0), CNPq (processo 300723/93-8) e CAPES (bolsa de doutorado para o primeiro autor).

8. Referências Bibliográficas

- [Aur95] - Aurrcoechea, C.; Campbell, A.; Hauw, L., "A Review of Quality of Services Architectures", ACM Multimedia Systems Journal, novembro de 1995.
- [Bel 91] - Belina, F.e outros, SDL with Applications from Protocol Specification, Prentice Hall International Ltd, 1991.
- [Cam 93] - Campbell, A.; Coulson, G. e Hutchim, D. "A Multimedia Enhanced Transport Service in a Quality of Service Architecture", Workshop on Network and Operating System Support for Audio and Video'93, Lancaster, England, novembro de 1993.
- [Che 95] - Chess, D. E outros, "Intinerant Agents for Mobile Computing", IBM Research Report RC 20010, IBM Reserarch Division, 1995.
- [Fur 94] - Furht, B. "Multimedia Systems: An Overview", IEEE Multimedia, pp. 47-59, Spring 1994.
- [Geo 95] - Geode - Manual de Referência, Verilog inc., França, 1995.
- [Haf 95] - Hafid, A. e Bochann, G. "An Approach to Quality of Service Management for Distributed Multimedia Applications", ICODP'95, Austrália, pp. 319-340, fevereiro de 1995.

- [ION 95] - IONA Technologies Inc., "ORBIX/CORBA - Distributed Object Technology", versão 1.3.1, fevereiro de 1995.
- [ITU 96] - ITU (International Telecommunication Union) Home Page, disponível na URL <http://www.itu.ch>, acessada em novembro de 1996.
- [Ker 94] - Kerhervé, B. e outros, "On Distributed Multimedia Presentational Applications: Functional and Computacional Architecture and QoS Negotiation", Proc. 4th Int. Workshop on Protocols for High-Speed Networks, Chapman & Hall, London, pp. 1-17, 1994.
- [Nah 95] - Nahrstedt, K. e Smith, J. "The QoS Broker", IEEE Multimedia, pp.53-67, Spring 1995.
- [Nwa 96] - Nwana, H., "Software Agents: An Overview Knowledge Engineering Review, vol. 11, No 3, pp. 1-40, setembro de 1996.
- [ODP 95] - Reference Model of Open Distributed Processing (ODP) -Part 3: Prescriptive Model, Int. Standard 10746-3, ITU-T Recommendation X.903, ITU-ISO, Genebra, 1995.
- [Rud 96] - Rudolph, E. e outros, "Tutorial on Message Sequence Charts", disponível na URL <http://www.win.tue.nl/win/cs/fm/sjouke/msc.html>, pego em outubro de 1996.
- [Vog 95] - Vogel, A. e Kerhervé, B. "Distributed Multimedia and QoS: Survey, IEEE Multimedia, Summer, pp. 10-18, 1995.