

Coordenação de Atividades em Ambientes de Aprendizagem Colaborativos

Carlos José M. Olguin¹, Alberto B. Raposo² e Ivan Luiz M. Ricarte¹

¹ Departamento de Engenharia da Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6101 – 13083-970 – Campinas – SP – Brasil
{olguin, ricarte}@dca.fee.unicamp.br

² Grupo de Computação Gráfica (Tecgraf)
Departamento de Informática (DI)
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
R. Marquês de S. Vicente, 225 – 22453-900 – Rio de Janeiro – RJ – Brasil
abraposo@tecgraf.puc-rio.br

Resumo. No contexto da aprendizagem colaborativa apoiada por computador, trocas de argumentações são essenciais para incrementar o nível de conhecimento dos membros de um grupo. O registro das discussões ocorridas no grupo é uma boa ferramenta para resgatar a evolução da aprendizagem. No entanto, para facilitar a organização desses registros e para que esses grupos de discussão se tornem mais produtivos é interessante que essas discussões não ocorram de forma caótica. Uma maneira de organizar um grupo de discussão é estabelecer, antecipadamente, requisitos e momentos adequados para o desenvolvimento das atividades de discussão através de um modelo de coordenação. Este trabalho propõe a modelagem das atividades de discussão de um grupo de estudos utilizando um modelo de coordenação de atividades. A partir da modelagem formal do sistema, usando mecanismos de coordenação baseados em Redes de Petri, o comportamento do ambiente pode ser simulado e analisado, o que permite antecipar possíveis problemas e ajuda a tornar as interações envolvendo os estudantes mais eficientes.

Palavras chave: Aprendizagem colaborativa apoiada por computador, Coordenação.

1. Introdução

No estudo de técnicas de aprendizagem colaborativa apoiada por computador alguns pesquisadores trabalham na construção de sistemas que possam monitorar a colaboração automaticamente. Barros e Verdejo definiram um método para calcular um conjunto de atributos que caracteriza o comportamento individual e do grupo. Utilizando contribuições textuais rotuladas e lógica nebulosa é calculado o grau de colaboração do grupo [Barros 1999]. Mühlbrock e Hoppe propuseram considerar as ações dos usuários em um ambiente de trabalho compartilhado como base para a análise qualitativa das atividades realizadas. Esta

abordagem consegue derivar descrições das atividades do grupo através de um método de reconhecimento de atividades [Mühlbrock 2001].

Delgado et al. definiram um modelo para monitoração das atividades realizadas dentro de um grupo de estudos. Nesse trabalho foi proposto que as ações e as interações monitoradas sejam armazenadas, como contribuições, em um “grafo de discussão” [Delgado 2001]. Através da análise desta informação por um tutor humano seria possível inferir o processo cognitivo usado pelo estudante durante a discussão. Dando continuidade, o objetivo do presente trabalho é modelar as atividades de discussão realizadas dentro de um grupo de estudos, utilizando o

modelo de coordenação baseado em Redes de Petri proposto por Raposo e Fuks [Raposo 2002] e que tem se mostrado adequado para a modelagem de aplicações de aprendizagem colaborativa [Ricarte 2002]. O modelo resultante é passível de simulações e análises que permitam a antecipação de problemas, além de possibilitar a verificação formal do modelo e de ajudar a tornar mais eficientes as interações dos estudantes durante o processo de discussão. Adicionalmente, a partir do modelo formal do sistema, podem ser implementados componentes de software para dar apoio à discussão do grupo.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta as diretrizes de uma metodologia de desenvolvimento de ambientes colaborativos de aprendizagem que dá especial atenção à coordenação das tarefas que são executadas no ambiente. A Seção 3 descreve o modelo adotado para capturar as discussões de grupo; a Seção 4 introduz aspectos do modelo de coordenação adotado; a Seção 5 apresenta mecanismos de coordenação baseados em Redes de Petri; e a seção final apresenta as conclusões do trabalho com ênfase nas direções a serem seguidas para dar continuidade a este trabalho.

2. Metodologia de desenvolvimento

Uma metodologia de desenvolvimento de ambientes colaborativos de aprendizagem que tenha especial preocupação pela coordenação das tarefas que serão executadas neste ambiente deve, além de definir os requisitos e objetivos pedagógicos associados ao ambiente, considerar as seguintes atividades:

- 1) *Identificação das tarefas a executar no ambiente e suas interdependências;*
- 2) *Identificação e modelagem dos mecanismos de coordenação entre essas tarefas;*
- 3) *Análise do modelo desenvolvido em (2) cujos resultados podem levar de volta a (1);*
- 4) *Quando o resultado da análise (3) for satisfatório, realizar o projeto e a construção do ambiente.*

Neste trabalho o ambiente considerado é a troca de mensagens em grupos de estudos (Seção 3) para discutir sobre um determinado assunto, com a intenção de aumentar o nível de conhecimento dos integrantes do grupo sobre o assunto.

A fim de analisar o modelo de interdependências o mesmo deve ser formalizado. Neste trabalho se decidiu utilizar Redes de Petri porque elas possuem uma teoria bem estabelecida que possibilita a realização de simulações. Como resultado das análises e simulações feitas, podem ser detectados problemas tais que seja necessário voltar à fase de identificação das atividades do ambiente. Caso o resultado destas análises for satisfatório, se procede com o projeto e construção do ambiente que implemente o modelo analisado. Esta construção pode ser facilitada com o uso de componentes de software que implementem mecanismos de coordenação para as interdependências entre as tarefas.

Neste trabalho concentraremos nossa atenção nas atividades (1), (2) e (3) da metodologia, uma vez que o nosso interesse no momento é o de modelar atividades, através do uso de um modelo que considera aspectos de coordenação entre as tarefas. Embora questões relativas ao projeto e construção do ambiente não façam parte do escopo deste trabalho, é a nossa intenção estudar futuramente estes aspectos, explorando o uso de componentes de software.

3. Modelo de discussões do grupo

O projetista de um ambiente computacional para aprendizagem colaborativa deve, antes de construir sua aplicação, trabalhar com modelos que capturem os requisitos associados à utilização do ambiente e as correspondentes tarefas que deverão ser realizadas através dele. Neste artigo, será considerado um modelo para um ambiente de troca de mensagens textuais em grupos de estudo [Delgado 2001], que foi um dos motivadores deste trabalho.

O modelo considerado parte do pressuposto que durante as atividades do grupo, os serviços colaborativos do ambiente são usados pelos participantes para gerar algum tipo de informação que outros podem comentar ou discutir usando os mesmos serviços. Toda informação ou comentário diretamente associado à discussão é definido como uma contribuição. A contribuição é o elemento básico utilizado para monitorar as atividades do grupo. O modelo permite a quantificação das contribuições relevantes e aceitas, e a manutenção de um *log* de discussão, representado por uma seqüência de contribuições.

Uma discussão é representada como uma seqüência de contribuições. As contribuições

podem ser: (1) aceitas ou rejeitadas; (2) substituídas ou reconsideradas por uma outra contribuição; e (3) dúvidas, perguntas e respostas a uma contribuição.

O registro das contribuições é organizado em um *grafo de discussão* (Figura 1). O grafo representa uma discussão sobre um assunto - no exemplo da figura o assunto discutido é “Java Interfaces”. A discussão pode se encontrar em um dos seguintes estados: em andamento, concluída, parada e adiada. O significado dos estados em andamento e adiada é obvio. Uma discussão parada significa que a decisão de interrompê-la foi tomada (isto é, o grupo percebeu que a discussão tinha tomado um caminho sem fim). Uma discussão concluída significa que a discussão finalizou e que existe um conjunto de contribuições que foram aceitas pelo grupo como sendo um conhecimento válido. Discussões podem, eventualmente, disparar outras discussões associadas onde nem todos os membros do grupo estão envolvidos. Por exemplo, uma discussão pode gerar algumas conclusões e resultados, mas alguns membros do grupo podem querer discutir um pouco mais a respeito; ou o grupo pode ter decidido se dividir

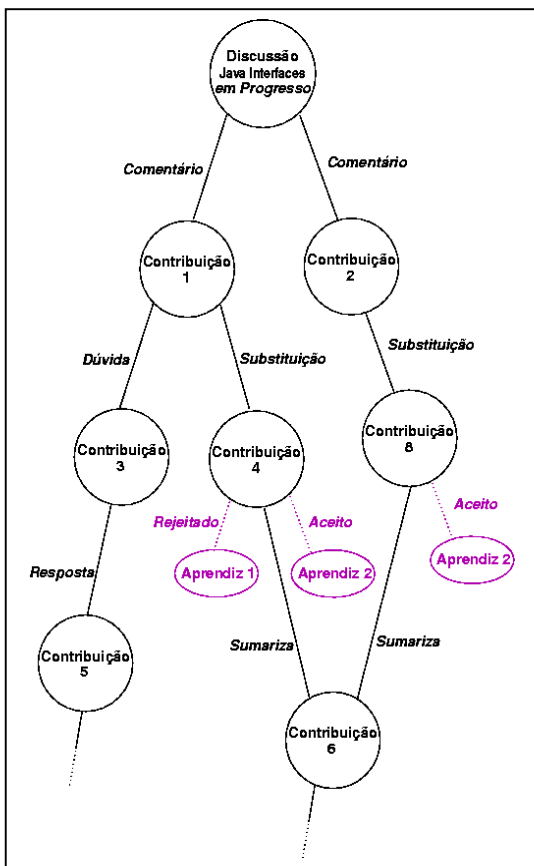


Figura 1. Grafo de discussão.

em subgrupos para aumentar o desempenho.

Os arcos do grafo têm associado um *tipo*. Este tipo expressa a relação que existe entre um nó e suas partes. Por exemplo, na Figura 1, *Contribuição 3* expressa uma dúvida sobre *Contribuição 1*; e *Contribuição 5* é a resposta àquela dúvida.

Em um dado momento, o grupo pode decidir produzir um sumário (parcial ou final) da discussão. Neste momento, todas as contribuições que têm suficiente aceitação por parte dos membros do grupo são agrupadas em uma contribuição especial que representa o sumário da discussão nesse momento. Como qualquer outra contribuição, o sumário é parte do grafo de discussão e está sujeito a todo tipo de análise (rejeição, aceitação, substituição, etc.). O conjunto dos sumários produzidos é o resultado da discussão e pode ser usado para inferir informação a respeito das crenças do grupo.

O processo de discussão do grupo envolve o conceito de “negociação do conhecimento” [Stahl 2002] que é um aspecto importante na construção e o gerenciamento do conhecimento gerado pelo grupo de estudos. Este processo abarca uma série de interações colaborativas entre os estudantes para, por exemplo, esclarecer o significado de alguns termos utilizados na colocação das contribuições, discutir alternativas, etc. [Stahl 2000].

A dinâmica do modelo de discussão apresentado nesta seção, onde um grupo de discussão pode decidir, por exemplo, dividir-se em subgrupos de discussão para aumentar o desempenho, ou onde o processo de aceitação/rejeição de uma contribuição envolve uma série de interações colaborativas, mostra a necessidade de coordenar as atividades realizadas. Na próxima seção é apresentado o modelo de coordenação de atividades utilizado neste trabalho.

4. Modelo de coordenação - interdependências

Uma atividade colaborativa, no nosso contexto, é definida com um conjunto interdependente de tarefas, executadas por diversos atores, de forma a alcançar um objetivo comum. No exemplo do nosso trabalho, os estudantes que fazem parte do grupo de discussão executam uma série de tarefas interdependentes (como por exemplo, efetuar contribuições, decidir sobre a aceitação

ou não delas como um conhecimento válido, etc.) com o objetivo de aumentar o nível de conhecimento sobre o assunto discutido.

O conceito de interdependência é fundamental na teoria de coordenação – se não houver nenhuma dependência entre tarefas executadas em um contexto colaborativo, não há nada o que coordenar [Malone 1994].

Uma abordagem separando as tarefas das suas interdependências é um passo dado no sentido de flexibilizar os mecanismos de coordenação. Uma das vantagens de separar as tarefas das suas interdependências é a possibilidade de alterar a política de coordenação simplesmente alterando os mecanismos de coordenação para as interdependências, sem a necessidade de alterar a essência do sistema colaborativo. Além disso, as interdependências e os seus mecanismos de coordenação podem ser reusados. É possível caracterizar diferentes tipos de interdependências e identificar os mecanismos de coordenação que os gerenciam, criando um conjunto de interdependências e respectivos mecanismos de coordenação que sejam capazes de atender uma grande gama de aplicações colaborativas [Malone 1994].

Nesta seção, é apresentado o conjunto genérico de interdependências entre tarefas de atividades colaborativas proposto por Raposo e Fuks [Raposo 2002].

4.1 Interdependências temporais básicas

As interdependências temporais estabelecem a ordem relativa de execução entre um par de tarefas. O conjunto de interdependências

temporais do modelo proposto está baseado em relações temporais definidas por Allen. Ele provou que existe um conjunto de relações primitivas, mutuamente exclusivas, que poderiam ser aplicadas aos intervalos de tempo (i.e., qualquer par de intervalos de tempo está relacionado por uma única relação) [Allen 1984].

Um intervalo de tempo é caracterizado por dois eventos que estão associados a instantes de tempo. O primeiro evento é o início do intervalo A, denotado aqui como i_a . O outro evento é o fim do mesmo intervalo, denotado como f_a , sempre sendo $i_a < f_a$. De acordo com Allen, o conjunto de sete relações primitivas, mostrado na Figura 2, pode manter informação temporal sobre qualquer par de intervalos de tempo A e B.

Baseado nas relações da Figura 2 é definido um conjunto de axiomas para criar uma lógica temporal. Por exemplo, há axiomas para provar a exclusão mútua e a completude das relações básicas e outros para definir relações de transitividade, como por exemplo, se A *during* B e B *before* C, então é inferido que A *before* C.

O fato das relações serem aplicadas sobre intervalos de tempo (e não sobre instantes de tempo) fez com que estas fossem adequadas para coordenar tarefas, isto porque tarefas geralmente não são operações instantâneas. A adaptação das primitivas de Allen para o contexto de atividades colaborativas considera que qualquer tarefa T ocupará algum tempo (de i para f) para ser executada.

A lógica temporal de Allen está definida em um contexto onde é essencial ter propriedades tais como a definição de um conjunto mínimo de relações básicas, a exclusão mútua entre estas relações e a possibilidade de tirar conclusões a

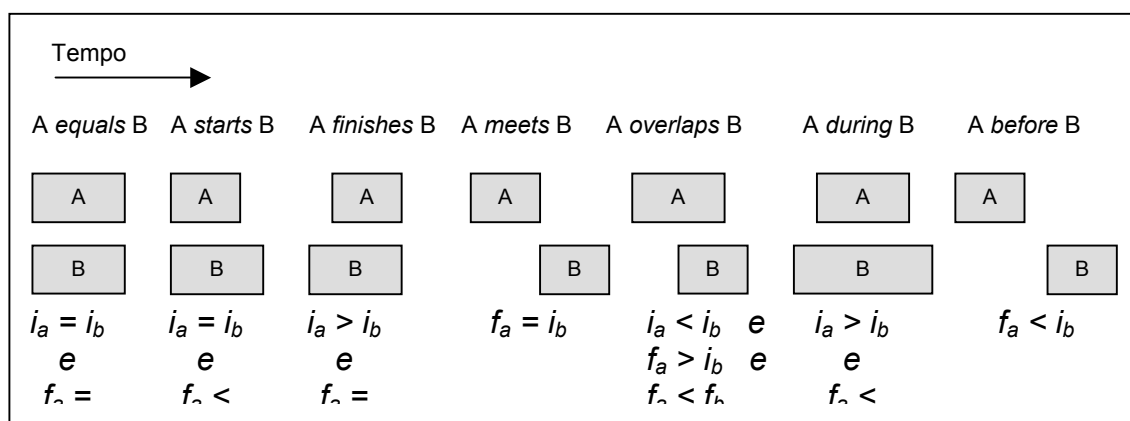


Figura 2. Conjunto de relações primitivas de Allen entre intervalos de tempo A e B.

partir delas. Porém, interdependências temporais entre tarefas colaborativas estão inseridas em um contexto diferente. O importante aqui é o gerenciamento das interdependências e o entendimento adequado da atividade colaborativa.

Uma limitação das relações de Allen é que elas são meramente descritivas, ou seja, elas não expressam relações causais ou funcionais entre os intervalos [Duda 1995]. Por exemplo, se as tarefas A e B estão relacionadas pela interdependência temporal *equals*, o que deveria fazer o mecanismo de coordenação quando a tarefa A está pronta para iniciar, mas não a tarefa B? Deveria bloquear a execução da tarefa A até que a tarefa B esteja pronta, ou deveria forçar o início da tarefa B para garantir que a interdependência será respeitada? Em uma situação diferente, se é dito que a tarefa A acontece antes da tarefa B, o que deveria ser feito se a tarefa B está pronta mas não a tarefa A? O mecanismo de coordenação deveria bloquear a tarefa B até o fim da tarefa A, ou deveria permitir a execução da tarefa B, enquanto bloqueia outras execuções da tarefa A (as quais violariam a relação)?

Por todas estas razões, foi necessário fazer algumas adaptações às relações básicas de Allen. Mais do que responder perguntas como as feitas acima, a meta das extensões propostas é oferecer um conjunto maior de possibilidades para criar mecanismos de coordenação que possam controlar muitas situações diferentes.

4.2 Interdependências ativas e passivas

A característica meramente descritiva das relações temporais de Allen permite diferentes interpretações de uma única interdependência. Por exemplo, considere que duas tarefas, T_a e T_b , estão relacionadas pela interdependência $T_a \text{ equals } T_b$. Esta construção estabelece que as duas tarefas especificadas devem ser executadas simultaneamente. No contexto de coordenação, esta interdependência pode ser interpretada de duas maneiras diferentes. No primeiro caso, chamada de interpretação ativa, esta relação expressa que o início de uma tarefa deveria iniciar a outra; e que o fim de uma das tarefas deveria finalizar a outra tarefa. Vamos considerar uma situação na qual uma tarefa é “iniciar uma sessão de discussão” e a outra tarefa é “registrar uma sessão de discussão em andamento”. Do

ponto de vista de coordenação, a interpretação ativa da relação simplesmente indica que a segunda tarefa (registro da sessão) deve acompanhar a execução da primeira tarefa. Porém, se houver um problema para continuar registrando a sessão, isto não a invalida.

A segunda interpretação possível para qualquer mecanismo de coordenação é chamada de interpretação passiva. Neste caso, o mecanismo de coordenação expressa um conjunto de condições que devem ser obedecidas para levar a cabo a atividade. Considerando o mesmo exemplo acima, este seria o caso se a tarefa de registrar a sessão devesse estar pronta antes do início da discussão. Assim, um problema para registrar a sessão atrasaria o início da discussão até o problema ser resolvido.

Para lidar com as interpretações ativas e passivas, dois operadores foram definidos: *enables* e *forces*. O operador *enables* representa a interpretação passiva, enquanto *forces* representa a ativa. Estas operações podem ser aplicadas nos momentos iniciais e finais de cada tarefa interdependente. Adicionalmente, estes pontos extremos têm dois estados, *ready* e *concluded*, indicando, respectivamente, que a tarefa está pronta para iniciar (ou terminar) e que já iniciou (ou terminou).

A fim de exemplificar as interpretações passivas e ativas das interdependências temporais são apresentados, a seguir, alguns exemplos de tarefas e suas respectivas interdependências (para ver o conjunto de interdependências temporais e suas respectivas interpretações, bem como outras interpretações e seus respectivos operadores consulte [Raposo 2002]).

Considere por exemplo duas tarefas, Trc (o registro de um comentário) e Tse (a solicitação de esclarecimentos) com pontos de início e fim i_{rc} , i_{se} , f_{rc} e f_{se} . Neste caso a interdependência associada às duas tarefas é $Trc \text{ before } Tse$ dado que o pedido de esclarecimentos a respeito de um comentário só pode ser feito uma vez que o comentário tenha sido registrado. A única interpretação que pode ser dada a esta interdependência é a passiva, isto é, $f_{rc} \text{ (concluded) enables } i_{se}$. Considerando somente os instantes iniciais e finais das tarefas não é possível dar uma interpretação ativa à interdependência porque esta interdependência impõe um intervalo de tempo entre as tarefas (se não houver um intervalo de tempo entre as tarefas, a interdependência entre elas seria *meets*).

Considere agora a situação em que dois estudantes, A e B, têm a tarefa de votar pela aceitação ou rejeição de uma contribuição. Denominaremos T_{va} a tarefa de votar associada ao estudante A, e T_{vb} a tarefa de votar associada ao estudante B, com pontos de início e fim i_{va} , i_{vb} , f_{va} e f_{vb} respectivamente. Imagine agora que no nosso ambiente o processo de votação se inicia de maneira síncrona. Neste caso, a interdependência associada às duas tarefas é T_{va} starts T_{vb} que pode ser estendida em várias interpretações.

Para o início simultâneo as seguintes interpretações podem ser feitas:

i_{va} (ready) enables i_{vb} AND i_{vb} (ready) enables i_{va} – esta declaração indica a situação passiva na qual as tarefas só iniciarão a sua execução quando ambas estiverem prontas (i.e., T_{vb} poderá iniciar somente quando T_{va} estiver pronta para começar, e vice-versa), mas nenhuma forçará a execução da outra.

i_{va} (ready) forces i_{vb} – nesta situação, quando T_{va} estiver pronta para iniciar, T_{vb} será forçada a começar, indicando uma interdependência ativa do tipo “mestre/escravo” (da mesma forma, T_{vb} poderia ser considerada a tarefa mestre se i_{vb} (ready) forces i_{va}).

i_{va} (ready) forces i_{vb} AND i_{vb} (ready) forces i_{va} – interdependência ativa sem mestre (o início de uma tarefa forçará o início da outra).

i_{va} (ready) forces i_{vb} AND i_{vb} (ready) enables i_{va} – T_{va} é a tarefa mestre, já que força o início da tarefa T_{vb} , mas T_{va} só iniciará quando T_{vb} estiver pronta.

4.3 Interdependências para o gerenciamento de recursos

Para complementar a especificação de tarefas colaborativas, três construções básicas associadas à disponibilidade de recursos foram definidas, a saber: compartilhamento, simultaneidade, e volatilidade [Raposo 2000].

Compartilhamento (*sharing*) representa uma situação na qual o uso de um número limitado de

recursos, expresso como um parâmetro, é requerido para a execução de várias tarefas. A dependência *sharing 1* representa a exclusão mútua, já que qualquer tarefa que use o recurso bloqueará todas as outras tarefas que quiserem usar aquele recurso.

Simultaneidade (*simultaneity*) representa a situação na qual o recurso especificado só está disponível quando pelo menos o número de tarefas especificado na dependência pede o recurso. Por exemplo, a dependência *simultaneity 2* pode ser usada para representar que uma sessão de *chat* não deve ser aberta quando solicitada por um único usuário.

Volatilidade (*volatility*) indica se um recurso, depois do seu uso, ainda está disponível para outras tarefas ou é consumido pela tarefa que o utilizou. Um parâmetro especifica quantas vezes o recurso pode ser usado antes de ficar indisponível para o sistema. Por exemplo, para garantir que cada participante vote apenas uma única vez pela aceitação ou não de uma contribuição, um recurso “cédula individual” necessária à tarefa “votar sobre contribuição” pode ter *volatility 1*.

Estas dependências podem ser combinadas para expressar relações mais complexas entre tarefas e recursos. Por exemplo, *sharing 2 volatility 3* expressa que um determinado recurso pode ser usado simultaneamente por até duas tarefas, mas depois do seu terceiro uso o recurso fica indisponível.

5. Mecanismos de coordenação baseados em Redes de Petri

Mecanismos de coordenação baseados em Redes de Petri foram desenvolvidos para coordenar o conjunto de interdependências apresentado anteriormente. A modelagem formal permite ao projetista antecipar e testar o comportamento do ambiente. Redes de Petri podem capturar as características principais de um ambiente colaborativo, tais como não-determinismo, concorrência e a sincronização de processos assíncronos. Elas suportam modelos a níveis de abstração diferentes e são adequadas para simulação e verificação formal.

Nesta proposta, o projeto de um ambiente de aprendizagem colaborativo é dividido em três níveis hierárquicos: *workflow*, coordenação, e execução (Figura 3). No nível de *workflow*, são estabelecidas as interdependências entre as tarefas atribuídas aos atores do ambiente (atividade 1 da metodologia de desenvolvimento apresentada na Seção 2). O nível de coordenação é construído a partir do nível de *workflow*, por meio da expansão das tarefas interdependentes e da inserção dos mecanismos de coordenação correspondentes (atividade 2 da metodologia de desenvolvimento). O modelo do ambiente é simulado e analisado neste nível (atividade 3). O nível de execução lida com a execução das tarefas no ambiente (atividade 4).

fluxo é dividido em dois caminhos paralelos, um que indica que a tarefa está em execução – *i(concluded)* – e outro que representa a interação com a execução da tarefa no sistema. A execução de tarefa é modelada através de uma transição com reserva de *tokens* (representado com a letra “R”) que é uma transição não-instantânea – os *tokens* são removidos do seu lugar de entrada quando a transição é disparada e somente algum tempo depois é acrescentado aos seus lugares de saída, representando desta forma a duração da tarefa.

Ao considerar duas tarefas relacionadas por interdependências, é necessário interconectar corretamente os lugares e as transições de ambos

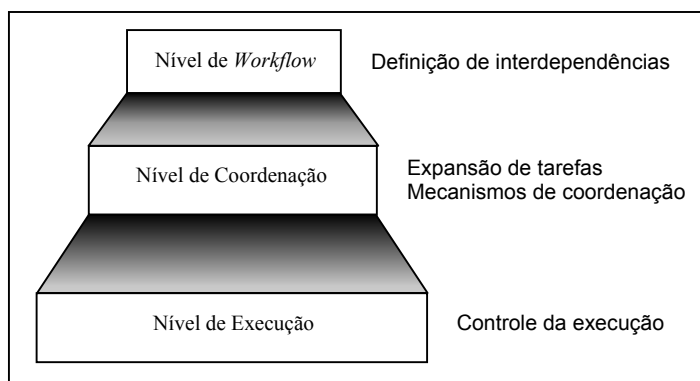


Figura 3. Níveis de projeto para os mecanismos de coordenação.

Durante a passagem do nível de *workflow* para o de coordenação, cada tarefa que tem uma interdependência com outra é expandida em uma sub-rede como a apresentada na Figura 4. Neste modelo, os eventos *i* e *f* (início e fim da tarefa) são representados como transições, enquanto os estados *ready* e *concluded* são representados como lugares conectados às respectivas transições. Depois de disparado o evento *i*, o

os modelos para criar os mecanismos de coordenação correspondentes. Para fazer isto, é preciso definir como mapear os operadores e parâmetros previamente definidos para os modelos de Redes de Petri. O mapeamento do *forces* e de outros operadores para o modelo de Redes de Petri são apresentados em [Raposo 2002].

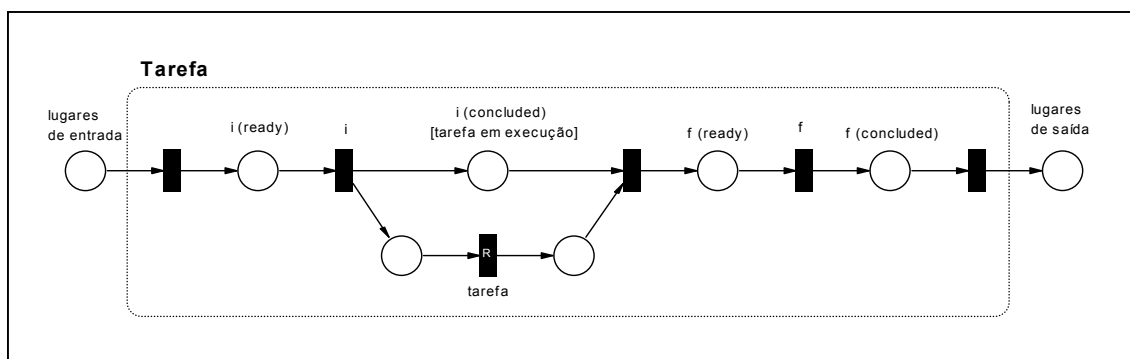


Figura 4. Representação em Redes de Petri de uma tarefa interdependente no nível de coordenação.

O mapeamento do operador *enables* é bastante simples. Este operador é modelado por um arco que sai do lugar que representa o primeiro operando para a transição que representa o segundo. Por exemplo, *ia (concluded) enables ib* é representado por um arco que sai do lugar *ia (concluded)* para a transição *ib*. É necessário também somar 1 ao peso do arco que chega ao primeiro operando, porque isto permite que aquele lugar habilite o fluxo normal da tarefa e o evento dado pelo segundo operando. Para ilustrar isto, considere a interdependência *Ta equals Tb* na interpretação passiva, i.e., *ia (ready) enables ib* AND *ib (ready) enables ia* AND *fa (ready) enables fb* AND *fb (ready) enables fa*. O mecanismo de coordenação para esta interdependência é ilustrado na Figura 5.

maneira de organizar um grupo de discussão é estabelecer requisitos e momentos adequados para o desenvolvimento das atividades de discussão através de um modelo de coordenação.

Neste trabalho foi apresentada uma proposta de utilização de um modelo de coordenação das atividades do grupo com a intenção de tornar as interações dos estudantes mais eficientes. O modelo utilizado, proposto por Raposo e Fuks [Raposo 2002], é baseado em um conjunto de interdependências entre tarefas e nos mecanismos de coordenação necessários para manipular tais interdependências. O conjunto de interdependências do modelo inclui dependências temporais (isto é, dependências que estabelecem a ordem relativa de execução entre um par de tarefas) e de gerenciamento de recursos (isto é, dependências que tratam da

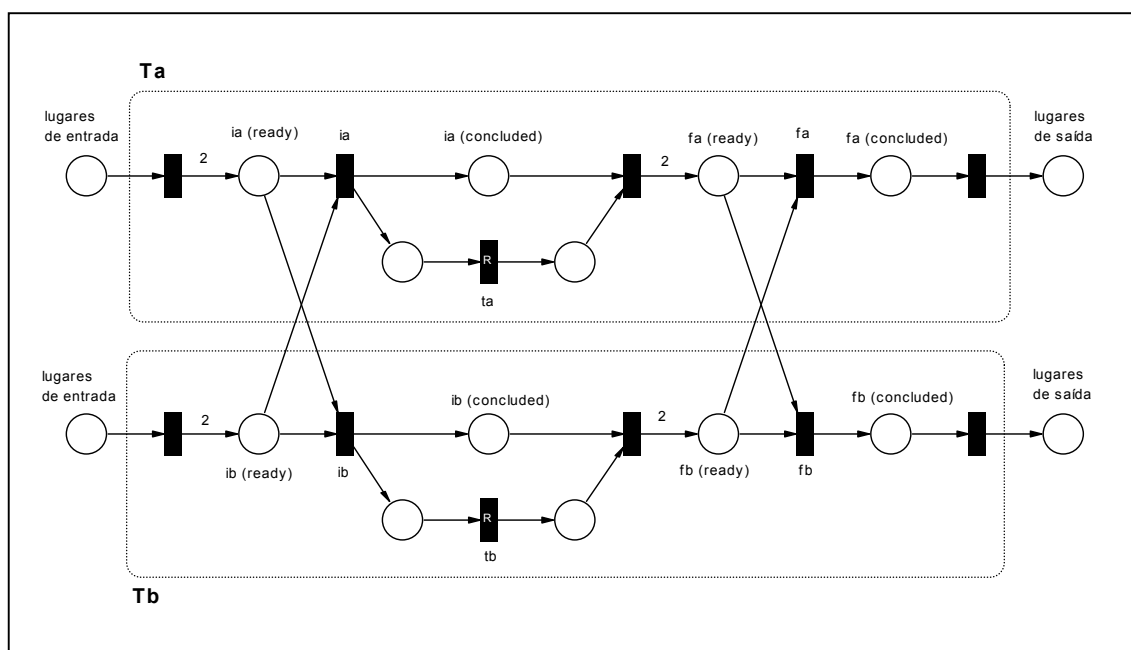


Figura 5. Mecanismo de coordenação para *Ta equals Tb*, interpretação passiva.

6. Conclusões

No contexto da aprendizagem colaborativa apoiada por computador, trocas de argumentações são essenciais para incrementar o nível de conhecimento dos membros de um grupo. O registro das discussões ocorridas no grupo é uma boa ferramenta para resgatar a evolução da aprendizagem no grupo. No entanto, para facilitar a organização desses registros e para que esses grupos de discussão se tornem mais produtivos, é interessante que essas discussões não ocorram de forma caótica. Uma

distribuição dos recursos entre as tarefas). O modelo proposto oferece um certo grau de flexibilidade através da separação entre as tarefas e suas interdependências (facilitando mudanças nas políticas de coordenação) e é adequado para tratar com alguns aspectos de interoperabilidade já que o conjunto de interdependências é genérico e a implementação dos mecanismos de coordenação pode ser realizada utilizando qualquer ferramenta.

Este modelo de coordenação é bastante adequado para representar as interações que acontecem entre os integrantes de um grupo de discussão. A

partir do modelo resultante podem ser derivados componentes de software para dar subsídios à discussão do grupo. A adoção de componentes de software permitirá padronizar a interação entre as tarefas e os mecanismos de coordenação associados a elas de uma forma independente da sua implementação. Uma abordagem inicial no sentido de derivar componentes de software que implementem os mecanismos de coordenação foi apresentada em [Raposo 2001].

Outra possibilidade a ser explorada é o uso de mecanismos de coordenação nebulosos, os quais podem trazer um grau de flexibilidade maior aos sistemas colaborativos para, por exemplo, adaptar o modelo de maneira a tratar com imprecisões próprias do raciocínio humano (isto é, permitir o início da execução de uma tarefa quando outra está “quase finalizada”).

Embora os mecanismos de coordenação apresentados representem adequadamente as interdependências entre as tarefas que compõem uma atividade colaborativa, o modelo resultante é estático. Esta característica do modelo impede que situações alternativas (i.e., atividades que podem ser realizadas através de conjuntos de tarefas diferentes) possam ser representadas. Neste sentido, a continuação deste trabalho está centrada na extensão dos mecanismos de coordenação apresentados de forma a torná-los mais flexíveis. Uma possibilidade seria a definição de novos operadores que permitam incorporar conceitos da lógica nebulosa para poder lidar com situações alternativas.

Agradecimentos. Carlos José Maria Olguín é professor do Departamento de Informática da Universidade Estadual de Maringá (DIN/UEM) e bolsista do programa PICDT da CAPES para a realização de estudos de doutorado na UNICAMP.

Referências

[Allen 1984] Allen, J. F. Towards a General Theory of Action and Time. *Artificial Intelligence*, vol. 23, pp. 123-154.

[Barros 1999] Barros, B. & Verdejo, M. An approach to analyze collaboration when shared structured workspaces are used for carrying out group learning processes. In: Proceedings of the 9th International Conference on Artificial Intelligence in Education – AIED’99, pp. 449-456, Lemans, France, July 1999.

[Delgado 2001] Delgado, A.L.N., Olguín, C.J.M. & Ricarte, I.L.M. Monitoring Learners Activities in a Collaborative Environment. In: Proceedings of the 7th International Workshop on Groupware – CRIWG 2001, pp. 148-152, Darmstadt, Germany, September 2001.

[Duda 1995] Duda, A. & Keramane, C. Structured Temporal Composition of Multimedia Data. In: Proceedings of the International Workshop on Multi-Media Database Management Systems – IW-MMDBMS, Blue Mountain Lake, USA, pp. 136-142, August 1995.

[Malone 1994] Malone, T. W. & Crowston, K. The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, vol. 26, no. 1, pp. 87-119.

[Mühlenbrock 2001] Mühlenbrock, M. & Hoppe, H. U. A Collaboration Monitor for Shared Workspaces. In: Proceedings of the 10th International Conference on Artificial Intelligence in Education – AIED’01, San Antonio, Texas, USA, May 2001.

[Raposo 2000] Raposo, A. B., Magalhães, L. P. & Ricarte, I. L. M. Petri Nets Based Coordination Mechanisms for Multi-Workflow Environments. *International Journal of Computer Systems Science & Engineering*, 15(5), pp. 315-326.

[Raposo 2001] Raposo, A. B., da Cruz, A. J. A., Adriano, C. M. & Magalhães, L. P. Coordination Components for Collaborative Virtual Environments. *Computers & Graphics*, vol. 25, no. 6, pp. 1025-1039.

[Raposo 2002] Raposo, A. B. & Fuks, H. Defining Task Interdependencies and Coordination Mechanisms for Collaborative Systems. In: Blay-Fornarino, M., Pinna-Dery, A. M., Schmidt, K. & Zaraté, P., Cooperative Systems Design (vol. 74 of Frontiers in Artificial Intelligence and Applications), pp. 88-103, IOS Press, Amsterdam, 2002.

[Ricarte 2002] Ricarte, I. L. M. & Raposo, A. B. Tasks Interdependencies in Collaborative Learning Activities: Specification and Modeling. In: Proceedings of the 5th IASTED International Conference on Computers and Advanced Technology in Education – CATE 2002, Cancun, Mexico, pp. 20-25, May 2002.

[Stahl 2000] Stahl, G. A Model of Collaborative Knowledge-Building. In: Proceedings of the Fourth International Conference of the Learning Sciences – ICLS 2000, pp. 70-77, Ann Arbor, USA, June 2000.

[Stahl 2002] Stahl, G., Appelt, W., Ruland, R., Hinrichs, E. & Woetzel, G. Supporting Knowledge Negotiation in Virtual Classrooms. Submetido ao CSCW 2002 – ACM 2002 Conference on Computer Supported Cooperative Work.