



Combinando Comunicação e Coordenação em Groupware

Marco Aurélio Gerosa
LES / PUC-Rio
R.M.S. Vicente, 225
22453-900, Rio de Janeiro,
Brazil
+55 21 2540-6915 x132
gerosa@inf.puc-rio.br

Alberto B. Raposo
Tecgraf / PUC-Rio
R.M.S. Vicente, 225
22453-900, Rio de Janeiro,
Brazil
+55 21 2512-5984
abraposo@tecgraf.puc-rio.br

Hugo Fuks
LES / PUC-Rio
R.M.S. Vicente, 225
22453-900, Rio de Janeiro,
Brazil
+55 21 3114-1500 x4304
hugo@inf.puc-rio.br

Carlos J.P. de Lucena
LES / PUC-Rio
R.M.S. Vicente, 225
22453-900, Rio de Janeiro,
Brazil
+55 21 2540-6915 x103
lucena@inf.puc-rio.br

RESUMO

Neste artigo, é apresentada uma proposta para articulação baseada em modelos de representação de comunicação e coordenação para atividades colaborativas apoiadas por groupware. Para a representação das fases de pré e pós-articulação, clichês de conversação são utilizados. Para a fase de coordenação, um modelo que separa as tarefas de suas interdependências é utilizado. O esquema de articulação é então aplicado a um exemplo de business-web.

Palavras-chave

Groupware, Articulação, Coordenação, Comunicação, Workflow, Business-Web.

ABSTRACT

In this paper, we present a proposal for the articulation based on communication and coordination representation models for collaborative activities supported by groupware. For the representation of pre- and post-articulation phases, conversation clichés are used. For the coordination phase, a model separating the tasks and their interdependencies is used. The articulation schema is then applied to a business-web example.

Keywords

Groupware, Articulation, Coordination, Communication, Workflow, Business-Web.

1. INTRODUÇÃO

Planejamento é essencial na realização colaborativa de tarefas complexas. O planejamento visa garantir que o objetivo final seja alcançado a partir das tarefas individuais, gerenciando a natureza distribuída do trabalho. Em CSCW (*Computer Supported Cooperative Work*), a noção de planejamento é realizada pelo trabalho de articulação. Sem o trabalho de articulação, há o risco dos participantes se envolverem em tarefas conflitantes ou repetitivas [19].

A articulação envolve a pré-articulação das tarefas, o gerenciamento do andamento das mesmas e a pós-articulação. A pré-articulação envolve as ações necessárias para preparar a colaboração, normalmente concluídas antes do trabalho colaborativo se iniciar: identificação dos objetivos, mapeamento destes objetivos em tarefas, seleção dos participantes, distribuição das tarefas entre eles, etc. A pós-articulação ocorre após o término

das mesmas, e envolve a avaliação e análise das tarefas realizadas e a documentação do processo de colaboração (memória do processo). O gerenciamento do andamento, ou coordenação, cuida das interdependências entre as tarefas, precisando ser renegociada de maneira quase contínua ao longo de todo o tempo [14].

Algumas atividades, principalmente as atividades colaborativas fracamente integradas [16] como as que acontecem nos chats e audio/videoconferências, são intimamente associadas com relações sociais e geralmente são satisfatoriamente coordenadas pelo protocolo social em vigor. Nestes casos, não há mecanismos de coordenação explícitos, ficando a articulação a cargo da habilidade dos participantes em mediar as interações (a coordenação é culturalmente estabelecida e fortemente dependente da percepção mútua). Mecanismo de coordenação, no contexto deste artigo, se refere a um dispositivo de software especializado, que interage com uma aplicação de software específica de forma a dar suporte ao trabalho de articulação [20].

Por outro lado, há um grande número de atividades em grupo (atividades colaborativas fortemente integradas) que requerem mecanismos de coordenação sofisticados para que tenham um suporte computacional eficaz. Nestes tipos de atividades, uma tarefa depende das outras para iniciar, para ser executada ou para ser finalizada.

Business-webs (b-webs), que são as redes formadas pelos produtores, fornecedores, provedores de serviços e de infraestrutura e consumidores ligados por canais digitais [24], envolvem diversas atividades colaborativas inter-relacionadas, gerando uma forte necessidade de suporte computacional para a coordenação. Neste contexto, este artigo apresenta um esquema de articulação baseado em clichês de conversação e num modelo de tarefas e interdependências. Como ambas representações são automatizadas e modelam a interação e o fluxo de trabalho, elas capturam o histórico da conversação e permitem registrar os eventos ao longo da atividade colaborativa. A partir deste registro pode se recuperar o contexto das decisões e o raciocínio que levou a elas, possibilitando rever decisões ou reaplicá-las em contextos similares.

É importante mencionar que este trabalho segue um modelo de colaboração baseado em comunicação, coordenação e cooperação (o modelo de colaboração 3C [6],[10]). De acordo com este modelo, para colaborar, as pessoas se comunicam. Durante esta comunicação, compromissos são gerados e negociados. Estes compromissos implicam em tarefas que serão necessárias para

concluir o trabalho. Estas tarefas são gerenciadas pela coordenação, que organiza o grupo e garante que as tarefas sejam realizadas na ordem correta, no tempo correto e cumprindo as restrições e objetivos impostos. Durante a cooperação, enquanto os membros do grupo operam em conjunto no espaço compartilhado, as tarefas são realizadas. Entretanto, enquanto trabalhando, surge a necessidade de renegociar e tomar decisões. Desta forma, uma nova rodada de comunicação é demandada, que por sua vez, irá modificar e gerar mais compromissos que necessitarão da coordenação para que as tarefas a serem realizadas na cooperação sejam reorganizadas. Isto mostra o aspecto cíclico da colaboração. O modelo de colaboração 3C é resumido na Figura 1.

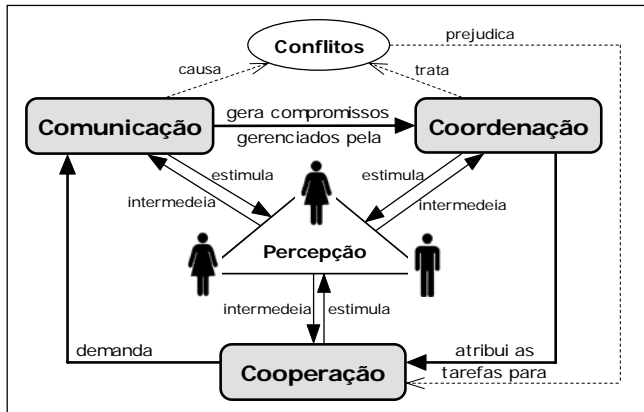


Figura 1. Modelo de colaboração 3C

Na seção seguinte, os modelos de representação são detalhados. Na Seção 3, estes modelos são aplicados em uma b-web. Na Seção 4, o esquema de articulação combinando os dois modelos de representação é discutido à luz de trabalhos relacionados.

2. MODELOS DE REPRESENTAÇÃO

Esta seção apresenta dois modelos de representação englobando todo o aspecto da articulação. Para as fases de pré e pós-articulação é proposto o uso de clichês de conversação para a negociação, estruturação e avaliação do trabalho colaborativo. Para a fase de coordenação, é proposto um modelo onde os compromissos gerados durante a conversação (pré-articulação) definem as atividades colaborativas, e mecanismos de coordenação são criados para gerenciar as interdependências entre as tarefas executadas pelos membros do grupo.

2.1 Clichês de Conversação

Compromissos resultam de uma conversação para ação [25], e representam um contrato informal com outras pessoas, indicando uma nova responsabilidade, uma obrigação, uma restrição ou uma decisão. A pessoa age de acordo com seus compromissos. Para uma melhor compreensão de nosso entendimento da noção de compromisso, consulte [8]. Há outras noções de compromissos na literatura de computação, como por exemplo em [2], [4], [12], [27].

Como os compromissos dão origem às tarefas do grupo, que são gerenciadas pela coordenação e executadas através da cooperação, a representação e gerenciamento dos compromissos

têm um papel importante no apoio à colaboração como um todo. Porém, como estes compromissos são provenientes da comunicação em linguagem natural, torna-se muito difícil capturá-los. Entretanto, alguns tipos de negociações, como as que envolvem um domínio conhecido, tendem a ser repetitivas e ter uma certa estrutura. Pode-se então identificar previamente partes das estruturas do discurso que se repetem – os clichês – e os reflexos delas nos compromissos de cada membro do grupo, armazenados em suas carteiras.

Clichês e atos de fala podem ser utilizados para auxiliar a captura das decisões e conhecimento gerado pelo grupo durante os processos argumentativos. Um clichê pode ser visto como uma máquina de estados que controla os eventos de diálogo que podem ser trocados entre os participantes do diálogo. Um determinado clichê restringe as possibilidades de caminhos pelos quais um diálogo pode navegar. Dessa forma é possível que se estabeleçam clichês característicos de diálogos realizados para determinados objetivos.

A estrutura de representação de diálogos chamada ACCORD, proposta em [9], pode ser utilizada para representar estes clichês através de máquinas de estado. Nesta representação, um diálogo pode ser visto como uma seqüência de eventos. Ao representar clichês que normalmente ocorrem numa determinada situação e as modificações que eles causam, pode-se identificá-los a partir do discurso dos participantes, refletindo as mudanças nas carteiras de compromissos, ou então guiar o discurso a partir de clichês pré-estabelecidos.

Cada participante tem uma carteira de compromissos que contém os compromissos por ele assumidos durante o diálogo. Esta carteira de compromissos é de domínio público, isto é, a carteira de compromissos de um determinado participante é do conhecimento de outro. O conteúdo da carteira de compromissos de cada participante é atualizado à medida que o diálogo progride. A única maneira de se alterar o conteúdo dessas carteiras é através do diálogo. Nas subseções seguintes serão abordados os construtores do esquema de diálogo do ACCORD [13].

2.1.1 Modificadores de Locução e Eventos de Diálogo

Uma locução consiste de uma ou mais sentenças qualificadas por um modificador de locução. Sua notação é *modificador de locução(sentença)*. As sentenças são construídas em linguagem proposicional que inclui negação, condicional, disjunção e conjunção de sentenças. Alguns modificadores de locução são:

Assertions, para ser lido “É o caso de <sentença>”, notação **asserts(sentença)**

Questions, para ser lido “É o caso de <sentença>?”, notação **questions(sentença)**

Withdraws, para ser lido “Eu não estou certo de <sentença>”, ou “Não me comprometo com <sentença>”, notação **withdraws(sentença)**

Cada locução afeta as carteiras de compromissos de modo próprio, conforme será visto adiante (subseção 2.1.3). Há outros quatro modificadores de locução que não são utilizados neste artigo. O ACCORD possui um cálculo para lidar com compromissos.

Os eventos de diálogo são representados por uma notação da forma $\langle P_1 \text{to} P_2, \text{Locução} \rangle$, onde P_1 e P_2 são participantes. Um diálogo é formado por uma seqüência de eventos de diálogo. Esta seqüência é armazenada em um registro de eventos.

2.1.2 Compromissos

Compromissos são armazenados em Carteiras de Compromissos, que são representadas da seguinte forma $\text{Participante}(C(\langle \text{Conjunto de sentenças} \rangle), D(\langle \text{Conjunto de sentenças} \rangle))$. Na área C ficam armazenados os compromissos do participante. Na área D ficam armazenados os não-compromissos desse mesmo participante. O não comprometimento com algo é diferente da não manifestação acerca daquilo.

Por exemplo, $(P_1(C(s_1, s_2)) \wedge P_2(C(s_1), D(s_3)))$ indica que a sentença 's1' está na área C dos participantes P_1 e P_2 , a sentença 's2' está na área C do participante P_1 e a sentença 's3' está na área D do participante P_2 . Isso significa que P_1 está comprometido com 's1' e 's2', enquanto P_2 está comprometido com 's1' e não está comprometido com 's3'.

Compromissos podem ser explícitos ou implícitos. Compromissos explícitos são aqueles gerados pelo processo de diálogo. Compromissos implícitos são aqueles inferidos pelo cálculo de compromissos. Graficamente, essas carteiras são representadas como mostrado na Figura 2.

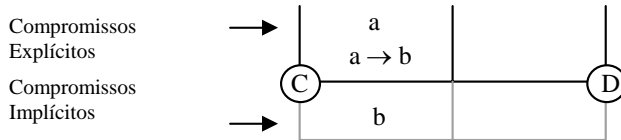


Figura 2. Representação gráfica de uma carteira de compromissos

2.1.3 Axiomas de Compromisso

Nesse trabalho, os diálogos são representados no seguinte formato:

$$\text{Pré} \rightarrow [P_1 \text{to} P_j, \text{Locução}]^n \text{Pós}$$

onde i e $j \in \{1, 2\}$, $i \neq j$; $n \geq 1$ e Pré e Pós são as condições antes e depois do diálogo. Pré e Pós são compromissos. Quando Pré ou Pós estão instanciados com *healthy*, o estado atual do diálogo é válido.

Axiomas de compromissos definem as mudanças nas carteiras de compromissos causadas pelos modificadores de locução. A seguir são apresentados os axiomas de compromisso para alguns modificadores de locução. " P_1 " e " P_2 " são participantes e " s " é uma <sentença>.

$$\text{healthy} \rightarrow [P_1 \text{to} P_2, \text{asserts}(s)] P_1(C(s)) \wedge P_2(C(s))$$

A partir de um estado válido, P_1 faz uma asserção da <sentença> " s ". Isto resulta na introdução desta <sentença> na área C da carteira de compromissos de ambos os participantes.

$$\text{healthy} \rightarrow [P_1 \text{to} P_2, \text{withdraws}(s)] P_1(D(s))$$

A partir de um estado válido, P_1 retira seu compromisso em relação a <sentença> " s ". Isto resulta na introdução desta <sentença> na área D da carteira de compromissos de P_1 .

$$P_1(C(s)) \rightarrow [P_1 \text{to} P_2, \text{withdraws}(s)] P_1(D(s))$$

Tendo P_1 o compromisso com a <sentença> " s ", ele retira seu compromisso com tal <sentença>. Isto resulta na retirada desta <sentença> da área C e sua inserção na área D da carteira de compromissos de P_1 .

$$\text{healthy} \rightarrow [P_1 \text{to} P_2, \text{questions}(s)] \text{healthy}$$

A partir de um estado válido, P_1 questiona P_2 em relação a <sentença> " s ". Este evento de diálogo não modifica as carteiras de compromissos de nenhum dos dois participantes.

Para mais detalhes sobre a teoria de clichês de conversação, consulte [9]. Neste artigo os clichês de conversação serão utilizados para as fases de pré e pós-articulação. Para a fase de coordenação das atividades será utilizado o modelo de representação de tarefas e interdependências descrito a seguir.

2.2 Modelo de Coordenação Baseado em Tarefas e Interdependências

Para a realização da coordenação é necessário ter uma definição clara dos conceitos de tarefas, atividades colaborativas e interdependências. No modelo adotado neste artigo, uma atividade colaborativa é definida como um conjunto de tarefas realizadas por vários membros de um grupo para atingir um objetivo comum (compromissos) [16]. Tarefas são os elementos constituintes das atividades colaborativas e estão conectadas por interdependências. Tarefas podem ser atômicas ou compostas de sub-tarefas. Um grupo de sub-tarefas pode ser considerado uma tarefa em um nível de abstração mais alto quando ele não possui interdependências com outras tarefas externas a este grupo. Isso garante a modelagem de atividades colaborativas em diferentes níveis de abstração.

Interdependência é um conceito chave na teoria da coordenação – se não há interdependências entre as tarefas a serem executadas, então não há nada a ser coordenado (note que isso não significa que não há a necessidade de pré- ou pós-articulação). A abordagem tarefa/interdependência é um passo em direção a oferecer flexibilidade aos mecanismos de coordenação, o que é crucial para o uso deste tipo de mecanismo. Essa abordagem é baseada em uma separação clara entre “o trabalho devotado à atividade de coordenação e o trabalho coordenado, i.e., o trabalho devotado à sua execução articulada no domínio alvo” [22].

Uma das vantagens da separação tarefa/interdependência é a possibilidade de se alterar políticas de coordenação simplesmente alterando os mecanismos de coordenação para as interdependências, sem a necessidade de se alterar o núcleo do sistema colaborativo. Além disso, as interdependências e seus mecanismos de coordenação podem ser reutilizados. É possível caracterizar diferentes tipos de interdependências e identificar os mecanismos de coordenação para gerenciá-los, criando um conjunto de interdependências e respectivos mecanismos de coordenação capazes de cobrir uma ampla gama de aplicações colaborativas [15]. Um exemplo de conjunto de mecanismos de coordenação que usam Redes de Petri para modelar as tarefas e o tratamento das interdependências é encontrado em [17].

A coordenação pode ocorrer em dois níveis — o nível de atividades (temporal) e o de objetos [7]. No nível temporal, a coordenação define o seqüenciamento de tarefas que constituem

uma atividade. No nível de objetos, a coordenação descreve como lidar com o acesso simultâneo ou seqüencial de múltiplos participantes a um mesmo conjunto de objetos de cooperação.

2.2.1 Interdependências Temporais

Interdependências temporais estabelecem a ordem de execução de um par de tarefas. O conjunto de interdependências temporais do modelo de coordenação proposto é historicamente baseado nas relações definidas por J. F. Allen [1]. Ele provou que existe um conjunto de sete relações primitivas e mutuamente exclusivas que podem ser aplicadas sobre intervalos de tempo. A adaptação das primitivas de Allen ao contexto das atividades colaborativas leva em consideração o fato de que uma tarefa T ocorre em um certo intervalo (de i até f).

Um intervalo de tempo é caracterizado por dois eventos, associados a dois instantes de tempo. O primeiro evento é o instante inicial de um intervalo A, denotado *ia*. O outro evento é o instante final do mesmo intervalo, denotado *fa*, sempre com $ia < fa$. As relações de Allen entre dois intervalos, A e B, são: **equals** ($ia=ib$ e $fa=fb$), **starts** ($ia=ib$ e $fa < fb$), **finishes** ($ia > ib$ e $fa=fb$), **before** ($fa < ib$), **meets** ($fa=ib$), **overlaps** ($ia < ib$, $ib < fa$ e $fa < fb$), **during** ($ia > ib$ e $fa < fb$).

Entretanto, as características descritivas das relações de Allen possibilitam diferentes interpretações para uma mesma interdependência. Suponha, por exemplo, que as tarefas A e B estejam associadas pela relação **equals**, que estabelece que ambas comecem e terminem no mesmo instante. Nessa situação, o que deve fazer o mecanismo de coordenação quando a tarefa A estiver pronta para começar, mas a B não? Ele deve bloquear a execução de A até a B ficar pronta para execução (interpretação passiva), ou deve forçar o início da tarefa B para garantir que a interdependência seja respeitada (interpretação ativa)? Em outra situação, se é dito que a tarefa A ocorre antes da B (A **before** B), o que deve ser feito quando a tarefa B está pronta, mas a tarefa A não? O mecanismo de coordenação deve bloquear a execução da tarefa B até o final da tarefa A, ou deve permitir a execução de B, bloqueando futuras execuções de A (que violariam a relação)? Por essas razões, foi necessário fazer algumas adaptações às relações básicas de Allen.

A primeira adaptação diz respeito às interpretações ativas e passivas das relações, como discutido acima, por meio de dois operadores: **enables** e **forces**. O operador **enables** representa a interpretação passiva, enquanto **forces** representa a ativa. Esses operadores podem ser aplicados sobre os instantes iniciais e finais de cada tarefa interdependente. Além disso, estes pontos extremos têm dois estados, *ready* e *concluded*, indicando respectivamente que a tarefa está pronta para começar (ou terminar) e que ela já começou (ou terminou). Esses estados são colocados no primeiro operando para indicar se ele vai habilitar ou forçar o segundo operando antes (*ready*) ou depois (*concluded*) de sua execução.

Considere, por exemplo, duas tarefas A e B, com instantes iniciais e finais *ia*, *ib*, *fa* e *fb*. A interdependência A **starts** B, que estabelece que ambas as tarefas devem começar simultaneamente, pode ser estendida em diferentes interpretações:

ia (ready) enables ib AND ib (ready) enables ia – essa sentença indica a situação passiva, na qual as tarefas vão iniciar apenas quando ambas estiverem prontas (i.e., B vai estar habilitada a

começar apenas quando A também estiver pronta, e vice-versa). Nenhuma das tarefas vai forçar a execução da outra.

ia (ready) forces ib – nessa situação, quando A está pronta para começar, B é forçada a começar, indicando uma relação mestre-escravo, de interpretação ativa. De maneira similar, B seria considerada mestre se *ib (ready) forces ia*.

ia (ready) forces ib AND ib (ready) forces ia – interdependência ativa sem mestre (o início de qualquer uma das tarefas força o início da outra).

Apesar dos operadores **enables** e **forces**, ainda existem situações que permanecem indefinidas. Um exemplo de tal situação ocorre em A **before** B. Depois que A e B tenham terminado, como deve proceder o mecanismo de coordenação se B estiver pronta para começar novamente? Ele deve permitir sua execução, uma vez que A já foi realizada (relação um para vários), ou deve fazer B esperar até que A seja executada novamente (relação um para um)? Dúvida semelhante surge em A **during** B, i.e., quantas vezes A pode ser executada durante uma única execução de B?

Para lidar com este tipo de situação, foi necessário incluir um parâmetro opcional para o operador **enables**. Esse parâmetro indica o número de vezes que uma condição (primeiro operando) habilita um evento (segundo operando).

Por exemplo, para definir que A **before** B permite um máximo de três execuções de B para cada execução de A, a seguinte sentença é usada: *fa (concluded) enables(3) ib*, indicando que, após cada execução de A, B pode ser executada até três vezes. Pode-se também definir que não há restrições quanto ao número de vezes que uma tarefa pode ser executada depois ou durante uma outra tarefa (seria o equivalente a definir o parâmetro com um valor infinito).

Para aumentar a flexibilidade do modelo, também foram criados os operadores **blocks** e **unblocks** que, respectivamente, desabilitam e reabilitam a execução de um evento (segundo operando) quando o estado do primeiro operando é alcançado. O uso destes operadores, por exemplo, possibilita uma nova interpretação de A **before** B:

ib (concluded) blocks ia – neste caso, há uma restrição na execução de A, que não poderá mais ser executada caso B já tenha iniciado sua execução. Não há restrição na execução de B (B não tem que esperar pela execução de A, como poderia acontecer com a situação dada por *fa (concluded) enables ib*).

2.2.2 Interdependências de Recursos

Interdependências relacionadas ao uso de recursos podem ser representadas como combinações de relações temporais. Por exemplo, se duas tarefas A e B não puderem usar o mesmo recurso simultaneamente, é possível definir uma dependência “não-paralelo” com as seguintes definições temporais: *ia (ready) blocks ib AND fa (concluded) unblocks ib AND ib (ready) blocks ia AND fb (concluded) unblocks ia*. Entretanto, além de ser sujeita a *deadlocks*, esta possibilidade ignora a noção de recurso, que é muito importante no contexto de workflows e atividades colaborativas. Por essa razão, não é suficiente tratar o problema de interdependências de tarefas simplesmente como um problema de lógica temporal. Além disso, considerando as dependências de gerenciamento de recurso independentemente das temporais, é criado um modelo de coordenação mais flexível,

permitindo ao projetista lidar com cada tipo de interdependência separadamente.

No modelo proposto, interdependências de gerenciamento de recurso são complementares às temporais e podem ser usadas em paralelo a elas. Este tipo de interdependência lida com a distribuição de recursos entre as tarefas. São definidas três interdependências básicas:

Sharing – um número limitado de recursos deve ser compartilhado entre várias tarefas.

Simultaneity – um recurso está disponível apenas se um certo número de tarefas o requisitarem simultaneamente. Representa, por exemplo, uma máquina que só pode ser usada por mais de um operador.

Volatility – indica se, depois do uso, o recurso fica disponível novamente. Por exemplo, uma impressora é um recurso não-volátil, enquanto uma folha de papel é volátil.

Cada um dos tipos de interdependência acima requer um parâmetro indicando o número de recursos compartilhados, o número de tarefas que devem requisitar o recurso simultaneamente e o número de vezes que um recurso pode ser utilizado (volatilidade).

Um mapeamento direto do modelo de coordenação apresentado para mecanismos de coordenação usando Redes de Petri, que englobam tanto as interdependências temporais quanto as de recursos foram apresentados em [16].

3. EXEMPLO EM UMA BUSINESS WEB

B-webs são definidas como “sistemas distintos de fornecedores, distribuidores, provedores de serviço e de infraestrutura e consumidores que utilizam a Internet como meio para suas comunicações e transações” [24]. Diferentes empresas estão cada vez mais re-agregando seus valores nestas b-webs, que estão se tornando uma das forças motoras da economia digital.

Uma b-web tem uma forte necessidade de persistência e gestão do histórico das conversações, ações e documentações produzidas, visto que ela envolve diversas empresas trabalhando em conjunto. Com isto, uma b-web é uma aplicação adequada para o esquema de articulação proposto. Ela fornece tarefas fortemente interdependentes e a fase de pré-articulação demanda normalmente conversações burocráticas, que são propícias para serem representadas por clichês.

B-webs são classificadas em cinco categorias: *agoras*, agregações, alianças, cadeia de valores e redes distribuídas. Olhando apenas para a cadeia de valores, sua organização principal é chamada de provedor de contexto, que estrutura e direciona a b-web para uma criação de valores altamente integrada [24]. Os outros participantes desta b-web cuidam da produção, entrega, atendimento ao consumidor, etc.

Cadeia de valores são diferenciadas por “produção rotineira” e “produção comercial”. Enquanto o primeiro tipo é centrado em produto e bens projetados para o mercado de massa e para eficiência de produção, o último dá suporte a soluções personalizadas onde as atividades são guiadas inteiramente pelas demandas que surgem, isto é, o consumidor é quem aciona o processo de criação de valores.

A Cisco Systems é um bom exemplo da cadeia de valores do tipo produção comercial. Usando a ferramenta de configuração do web site da Cisco, o consumidor é guiado na preparação de seu pedido, enquanto todo tipo de desconto e serviços são oferecidos. Somente depois de vender os bens é que a Cisco os produz. Na realidade, ao invés de produzir o bem, a Cisco coordena o processo de produção feito por seus parceiros.

A Cisco tem o papel de coordenador nesta cadeia de valores do tipo produção comercial. Usando a ferramenta de configuração disponível, os consumidores comunicam seus pedidos, disparando o ciclo de cooperação entre produtores, montadores, distribuidores, fornecedores e canais de venda. O objeto de cooperação é o produto final, seja ele um computador ou solução que será entregue para o consumidor. A Figura 3 ilustra este processo [11].

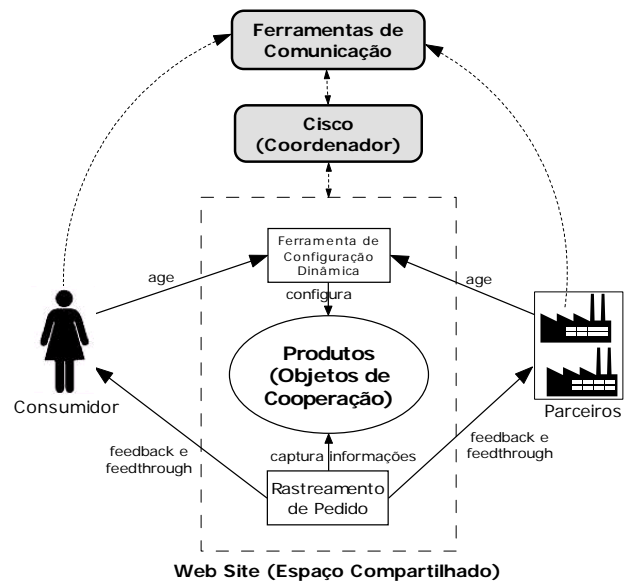


Figura 3. Cadeia de valores da Cisco.

Do ponto de vista da coordenação, é possível modelar o workflow da b-web usando o modelo de tarefas/interdependências apresentado anteriormente. Um exemplo de uma transação numa cadeia de valores hipotética é apresentado, envolvendo quatro participantes: um consumidor, o provedor de contexto, um produtor e um distribuidor. O workflow desta situação é apresentado na Figura 4. Nesta figura, as tarefas são descritas em ovais e suas relações em hexágonos. Setas pontilhadas indicam interdependências e setas normais, transições do workflow. A palavra *or* no workflow indica caminhos alternativos (apenas um deles é seguido). A ausência de *or* indica caminhos paralelos. As letras *a* e *b* próximas às tarefas inter-relacionadas são utilizadas para identificá-las nas diretivas contidas nos hexágonos.

Neste workflow, o processo é iniciado pelo consumidor que faz o pedido do produto no web site do provedor de contexto. Esta tarefa inicia o workflow do provedor de contexto (o final da tarefa do consumidor força o início do workflow do provedor de contexto). O consumidor espera então o produto ou cancela o pedido. Entretanto, o pedido só pode ser cancelado se o produto ainda não foi entregue. Isto define a relação entre as tarefas de *entrega de produto* (no workflow do provedor de contexto) e a de

cancelamento de ordem (no workflow do consumidor). A relação estabelece que o início da entrega bloqueia o cancelamento do pedido.

Depois de recebido o pedido, o provedor de contexto aciona duas tarefas paralelas: *confirma o pagamento* e *solicita o produto*. Esta tarefa inicia o workflow do produtor (*fa(concluded) forces ib*). Depois da confirmação de pagamento e do pedido, o provedor de contexto fica esperando até que o produto esteja pronto. Isto determina uma relação: quando o produtor terminar o produto, a tarefa *organiza entrega* no provedor de contexto é habilitada. Não utilizamos o operador *forces* porque é possível que o produto fique pronto antes da confirmação de pagamento, e neste caso, a entrega deve esperar.

Quando o provedor de contexto estiver pronto para organizar a entrega, ele inicia o workflow do distribuidor, que tem uma relação estreita com o resto do workflow do provedor de contexto. Esta relação é expressa em termos da relação *equals*: (*ia(ready) enables ib AND fa(ready) enables fb*), significando que as tarefas ocorrem simultaneamente no distribuidor e no provedor de contexto.

Este exemplo utilizou apenas relações de interdependências temporais, porém interdependências de recursos poderiam também ser utilizadas. Por exemplo, o provedor de contexto poderia ter um estoque, o que definiria uma rota alternativa ao contato com o produtor. A tarefa de retirar os produtos do estoque teria uma dependência volátil indicando que o estoque eventualmente acaba.

A fase de pré-articulação serve como entrada para o workflow descrito acima. A pré-articulação é parte da tarefa do consumidor *customiza produto*. Durante a execução desta tarefa, o consumidor interage com o web site da companhia e possivelmente com um operador. Durante esta interação, tanto o consumidor quanto a companhia assumem compromissos e, ao final da interação, estes compromissos definirão a configuração do produto e a data de

entrega. Para ilustrar, considere a seguinte situação.

O consumidor entra no web site e se registra. Para se registrar, aceita os termos e condições da companhia. Em seguida, entra na ferramenta de configuração dinâmica para pedir um produto e escolhe o produto número AB123. A ferramenta de configuração dinâmica oferece opções de configuração. O consumidor escolhe azul para ser a cor do produto e 220V para a voltagem. Em seguida, contacta um operador do site para negociar a data de entrega. O consumidor necessita do equipamento em no máximo três dias. O operador diz que para a cor azul isto não será possível. Só será possível na cor verde. O consumidor aceita esta cor, checka o preço e ordena o produto.

O clichê de conversão acima é representado pela seguinte seqüência de eventos de diálogo e carteiras de compromissos.

I. Diálogo de registro

healthy → [COMPANHIAtoCONSUMIDOR, **questions(t)**]
 [CONSUMIDORtoCOMPANHIA, **asserts(t)**]
 CONSUMIDOR (C(t)) ∧ COMPANHIA (C(t))

II. Diálogo de configuração

CONSUMIDOR (C(t)) ∧ COMPANHIA (C(t)) →
 [CONSUMIDORtoCOMPANHIA, **questions(p)**]
 [COMPANHIAtoCONSUMIDOR, **asserts(p)**]
 [CONSUMIDORtoCOMPANHIA, **questions(a)**]
 [COMPANHIAtoCONSUMIDOR, **asserts(a)**]
 [CONSUMIDORtoCOMPANHIA, **questions(v)**]
 [COMPANHIAtoCONSUMIDOR, **asserts(v)**]
 CONSUMIDOR (C(t,p,a,v)) ∧
 COMPANHIA (C(t,p,a,v))

III. Diálogo da negociação da data de entrega

CONSUMIDOR (C(t,p,a,v)) ∧ COMPANHIA (C(t,p,a,v)) →
 [CONSUMIDORtoCOMPANHIA, **questions(d)**]

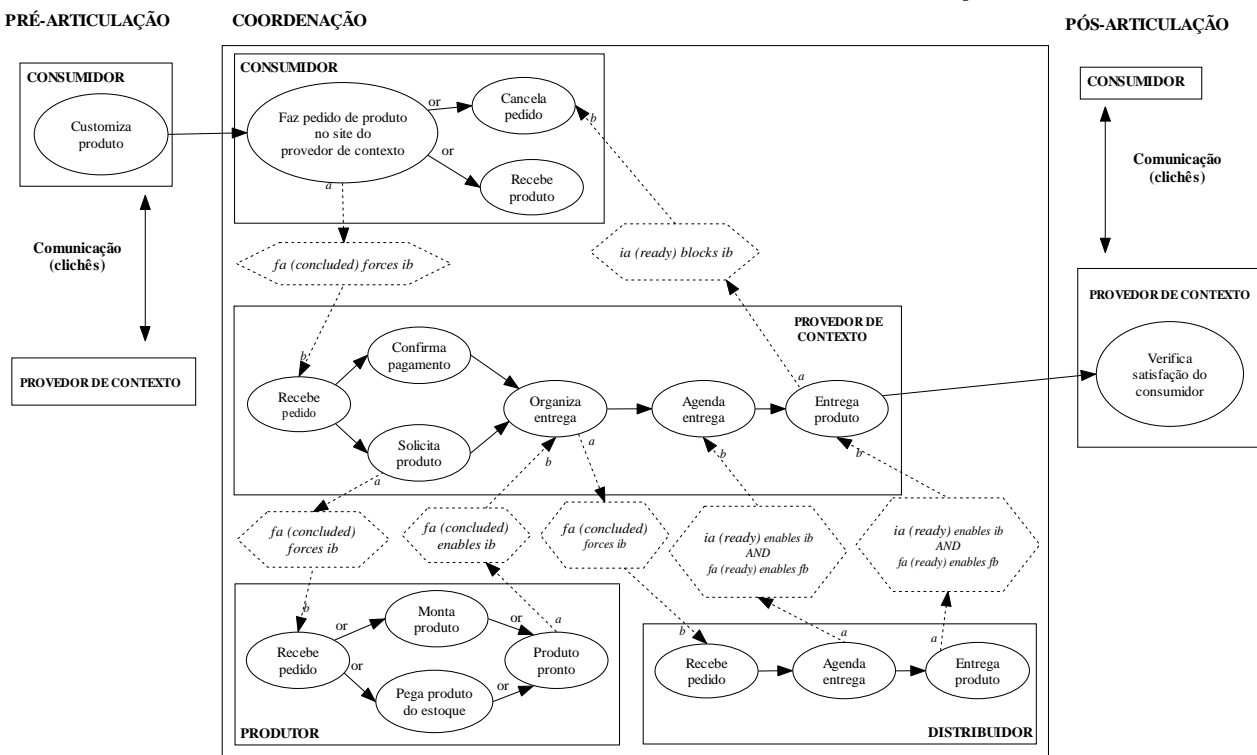


Figura 4. Workflow de uma transação típica numa cadeia de valores.

[COMPANHIAtoCONSUMIDOR, **withdraws(a)**]
 [CONSUMIDORtoCOMPANHIA, **withdraws(a)**]
 [COMPANHIAtoCONSUMIDOR, **questions(d)**]
 [CONSUMIDORtoCOMPANHIA, **asserts(e)**]
 [COMPANHIAtoCONSUMIDOR, **asserts(d)**]
 CONSUMIDOR (C(t,p,v,e,d),D(a)) ^
 COMPANHIA (C(t,p,v,e,d),D(a))

IV. Diálogo de finalização do pedido

CONSUMIDOR (C(t,p,v,e,d),D(a)) ^
 COMPANHIA (C(t,p,v,e,d),D(a)) →
 [COMPANHIAtoCONSUMIDOR, **questions(y)**]
 [CONSUMIDORtoCOMPANHIA, **asserts(y)**]
 CONSUMIDOR (C(t,p,v,e,d,y),D(a)) ^
 COMPANHIA (C(t,p,v,e,d,y),D(a))

Legenda:

t – respeitar os termos e condições da companhia
 p – produzir o produto AB123
 a – produzir o produto na cor azul
 v – produzir o produto em 220V
 d – entregar o produto em três dias
 e – produzir o produto em verde
 y – pagar pelo produto

Ao final do diálogo o consumidor está comprometido em respeitar os termos e condições da companhia e pagar pelo produto. A companhia está comprometida a respeitar seus termos e condições, e a produzir o produto em 220V, na cor verde e entregá-lo em três dias pelo preço estipulado. O compromisso de ter o produto na cor azul foi declinado por ambos (*withdraws a*). Os compromissos servem de entrada para o workflow descrito anteriormente, configurando e definindo suas tarefas. Também é possível definir clichês de conversação para a fase de pós-articulação, como por exemplo, em um diálogo entre a companhia e o consumidor para avaliar o grau de satisfação com o produto.

Vale reforçar que este exemplo lida com uma pequena parte de todo o modelo da b-web. Não enfatizamos todos os aspectos do cenário modelado (como por exemplo, as conseqüências do cancelamento, como reembolso, devolução, etc.). O principal objetivo aqui foi mostrar como os modelos de representação podem ser utilizados em uma situação prática.

Uma vantagem de utilizar estes modelos de representação para modelar uma b-web é que, através de uma análise matemática formal (como por exemplo, usando Redes de Petri no caso do workflow), é possível antecipar e testar o comportamento de ambientes interorganizacionais antes de iniciar sua implementação. Das primitivas do modelo de coordenação, também é possível definir um mapeamento direto para construir mecanismos de coordenação, como mostrado em [16]. No caso dos clichês de conversação é possível utilizar uma máquina de inferência, baseada num cálculo de compromissos para operar nas carteiras dos participantes, como mostrado em [9].

Com o uso deste esquema de articulação, a captura de informação proveniente das tomadas de decisões não fica restrita à documentação produzida pelos participantes, que constitui o

conhecimento formal. O conhecimento dito informal, isto é, idéias, fatos, questões, pontos de vista, conversas, discussões, decisões, etc. que ocorrem durante o processo e acabam por defini-lo, é registrado de forma organizada e estruturada. Isto possibilita recuperar o histórico da discussão e o contexto em que as decisões foram tomadas. Pode-se então investigar o raciocínio que levou ao projeto de um determinado artefato (*design rationale*) e averiguar posteriormente, em um novo contexto, se os motivos pelos quais as decisões de projeto foram tomadas continuam sendo válidas.

4. O ESQUEMA DE ARTICULAÇÃO

O esquema de articulação apresentado é ilustrado na Figura 5. Na fase de pré-articulação, clichês de conversação são utilizados para gerar os compromissos correspondentes às definições e alocações das tarefas e outros aspectos necessários na negociação da atividade colaborativa. Estes compromissos definem as tarefas do workflow, que coordena as interdependências entre as tarefas. Finalmente, a execução das atividades leva a uma fase de pós-articulação, onde os usuários podem documentar e inspecionar o processo, avaliar a qualidade do produto final, e realizar outras avaliações que podem modificar subseqüentes atividades.

Uma das características importantes deste esquema é a combinação de funcionalidades de diferentes abordagens de articulação: as baseadas em comunicação, representadas aqui pelos clichês de conversação, e aquelas apoiadas em mecanismos explícitos de coordenação. O esquema de articulação proposto é discutido nas subseções seguintes à luz de trabalhos relacionados.

4.1 Dimensões de Articulação

Carstensen e Nielsen, num estudo comparando coordenação oral e baseada em artefatos, definem algumas dimensões que caracterizam abordagens de articulação [3]. Com base nestas dimensões, é possível verificar que o esquema de articulação apresentado neste artigo combina vantagens de ambos tipos de coordenação.

- **Grau de automação:** é possível implementar computacionalmente o esquema de articulação proposto.
- **Persistência:** Todas as interações são registradas e armazenadas neste esquema de articulação. Esta é a característica que torna o esquema de articulação particularmente adequado à gestão do conhecimento.
- **Suporte a coordenação dedicado e não-exclusivo:** as fases de pré e pós-articulação envolvem tarefas que servem tanto para o trabalho quanto para a coordenação, já que são fortemente relacionadas com a comunicação (suporte a coordenação não-exclusivo). Na fase de coordenação, as tarefas de trabalho são completamente separadas das tarefas de coordenação (suporte à coordenação dedicado). Na realidade, esta separação é a principal característica do modelo de tarefas/interdependências.
- **Referência direta e indireta:** o clichê de conversação provê suporte para referencia: direta (dizer a alguém o que fazer) e indireta (prover informação para acessar o que fazer). Isto também está representado nos clichês pelos compromissos explícitos e implícitos. O modelo de coordenação, por outro lado, apresenta apenas referência direta, visto que ele explicitamente gerencia as interdependências entre as tarefas.

- **Dinâmico e estático:** as fases de pré e pós-articulação baseadas na comunicação tendem a ser dinâmicas, no sentido de que elas refletem as mudanças no mundo ao redor. O modelo de coordenação é essencialmente estático.
- **Acoplamento:** as fases de pré- e pós-articulação são desacopladas do campo de trabalho, no sentido de que não há uma conexão direta entre uma mudança de estado no meio de coordenação e uma mudança de estado no trabalho em si. Os mecanismos de coordenação, por outro lado, estão acoplados ao campo de trabalho (tarefas), no sentido de que suas mudanças de estados afetam as tarefas (por exemplo, por meio do operador **forces**).
- **Redução da carga de trabalho de coordenação:** o esquema de articulação aumenta a carga de coordenação durante a pré e pós-articulação, visto que é utilizado um modelo de conversação estruturado. O modelo de tarefas/interdependências, por outro lado, reduz a carga de coordenação por fornecer mecanismos de coordenação para regular a execução das tarefas.
- **Flexibilidade:** Esta dimensão será vista em mais detalhes na próxima subseção.

para os usuários, que ficam extremamente dependentes da percepção mútua. As críticas a esta abordagem é que a carga de coordenação é aumentada, visto que os usuários devem lidar com as complexidades da articulação das tarefas.

Há diferentes tipos de atividades colaborativas. Como mencionado anteriormente, atividades colaborativas altamente integradas necessitam de mecanismos de coordenação para regular as interações. Por outro lado, a carga de trabalho adicional imposta pela coordenação e a flexibilidade limitada imposta por estas abordagens são inadequadas para atividades colaborativas fracamente integradas, que são mais propensas às abordagens baseadas na percepção mútua e no protocolo social.

A despeito desta discussão, há uma tendência em conciliar ambas idéias, baseada no argumento de que os dois tipos de atividades aparecem inevitavelmente misturadas no mundo real [21]. Além disto, também é argumentado que devem ser providas facilidades para que os usuários possam interpretar e explorar padrões pré-definidos de coordenação, decidindo utilizar, alterar ou rejeitá-los [18]

Deiters et al., num estudo sobre flexibilidade em sistemas de workflow, definiu algumas dimensões de flexibilidade e soluções propostas para atingi-las [5]. Das suas dimensões de flexibilidade,

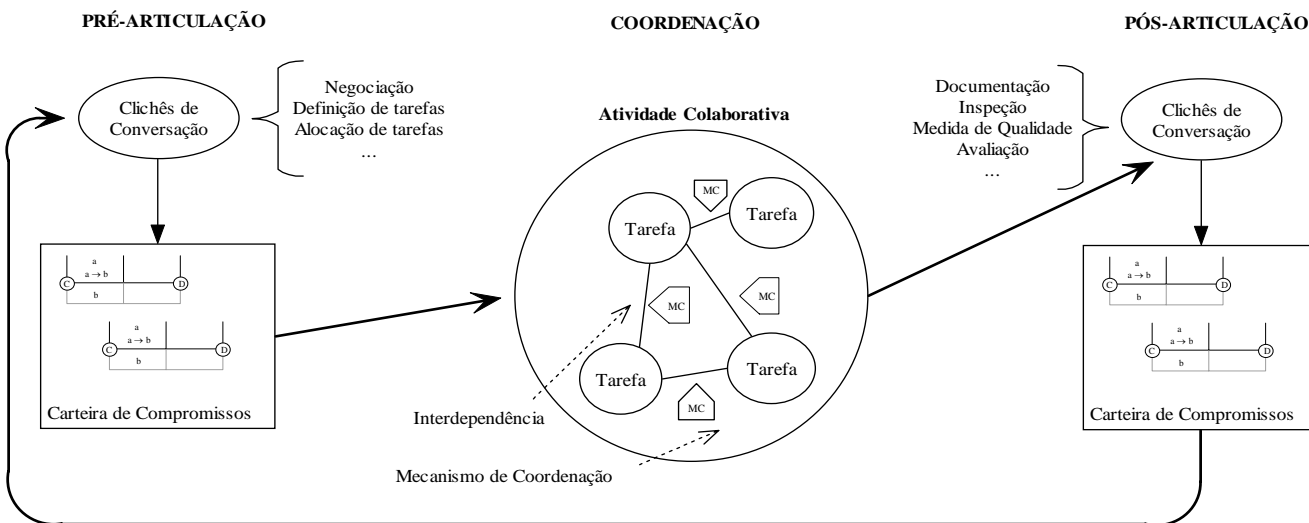


Figura 5. O esquema da articulação

4.2 Flexibilidade na Articulação

A necessidade de mecanismos de coordenação para regular as interações em sistemas colaborativos tem sido o alvo de uma calorosa discussão [23], [26]. De um lado, há modelos normativos que tentam regular a coordenação restringindo as interações entre os participantes e suas tarefas. Um sistema clássico que segue esta abordagem é o The Coordinator [25]. As críticas a estas abordagens normativas podem ser superficialmente resumidas pela constatação de que seus protocolos rígidos se aplicam apenas a cenários muito específicos, limitando a flexibilidade nos sistemas colaborativos. Eventualmente há situações não previstas pelos protocolos especificados, restringindo a aplicação dos mecanismos definidos.

Do lado oposto, há aqueles que alegam que sistemas colaborativos devem levar a flexibilidade ao extremo, deixando a articulação

há duas que se aplicam ao esquema de articulação proposto neste artigo.

- **Flexibilidade de processo:** relacionada a situações onde exceções ocorrem, e também quando há partes das atividades que não podem ser precisamente definidas a priori. Com relação ao esquema de articulação proposto, o fato de ter fases de pré e pós-articulação baseadas em comunicação atenua a rigidez da fase de coordenação, pois exceções e situações não definidas podem ser renegociadas e reconsideradas numa próxima rodada de coordenação.
- **Flexibilidade de alocação de tarefas:** é relacionada com a realocação de tarefas em tempo de execução. Deiters et al. [5] citou que um dos requisitos para flexibilidade na alocação de tarefas é a negociação da designação de papéis e decisões. Além disto, a própria negociação pode ser modelada como um

workflow. As fases de pré e pós-articulação baseada em clichês são um meio adequado para esta negociação.

O esquema de articulação proposto neste artigo é mais relacionado com atividades colaborativas altamente integradas, como as realizadas nas b-webs. Entretanto, o esquema possui algum grau de flexibilidade representado pela separação entre tarefas e interdependências no modelo de coordenação e pelo uso de comunicação durante a pré e pós-articulação. Comunicação, mesmo na forma de clichês, é mais flexível que a coordenação na forma de workflows.

5. CONCLUSÃO

Neste artigo foi proposto um esquema de articulação baseado em dois modelos de representação: clichês de conversação e modelo de tarefas/interdependências. Com relação ao modelo 3C, este esquema é um passo na direção de estabelecer um link implementável entre os aspectos estruturais da comunicação e da coordenação. Este esquema de articulação tem a vantagem de ser totalmente automatizado, podendo capturar e organizar o registro da interação, o que é importante para a gestão do conhecimento.

E-business, exemplificado aqui por uma b-web, capturam o lado estruturado da comunicação e coordenação, se tornando um campo de aplicação adequado ao esquema de articulação.

Considerando comunicação, ainda há espaço para a conversação desestruturada que ocorre no espaço compartilhado. O mesmo raciocínio se aplica à coordenação, visto que o workflow apenas captura a lado estruturado da coordenação.

Apesar de não ter sido explicitamente discutido neste artigo, o esquema de articulação não desconsidera o papel da percepção. A percepção ocupa um papel central no modelo de colaboração 3C [10]. Todo evento ocorrido durante a comunicação, coordenação e cooperação gera informações de percepção, que intermedeia todo o processo de colaboração. Percepção tem um efeito direto nas fases de pré e pós-articulação, que são mais dinâmicas, no sentido de refletir as alterações no mundo.

6. AGRADECIMENTOS

Os autores são financiados por bolsas individuais do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq): Carlos J.P. de Lucena nº 300031/92-0, Hugo Fuks nº 303055/ 02-2, Alberto B. Raposo nº 305015/02-8 e Marco A. Gerosa nº 140103/02-3. Agradecemos também ao prof. Marcelo Gattass, coordenador do Laboratório de Computação Gráfica da PUC-Rio.

7. REFERÊNCIAS

- [1] Allen, J. F. Towards a General Theory of Action and Time. *Artificial Intelligence*, 23, 1984, 123-154.
- [2] Bond, A. H. A Computational Model for Organizations of Cooperating Intelligent Agents. *Proceedings of Conference on Office Information Systems, SIGOIS Bulletin*, 11(2-3), 1990, 21-30.
- [3] Carstensen, P. H., and Nielsen, M. Characterizing Modes of Coordination: A comparison between oral and artifact based coordination. *Proceedings of GROUP'01 (Boulder CO, 2001)*, ACM Press, 81-90.
- [4] Cohen, R.P., and Levesque, J. H. Persistence, Intention and Commitment. In Cohen, P. R., and Perrault, C. R. (Eds.), *Formal Theories of Communication*, 171-203, Report n° CSLI-87-69, 1987.
- [5] Deiters, W., Goesmann, T., and Löffeler, T. Flexibility in workflow management dimensions and solutions. *International Journal of Computer Systems Science & Engineering* 15(5), 2000, 303-313.
- [6] Ellis, C.A., Gibbs, S.J., and Rein, G.L. Groupware - Some Issues and Experiences. *Comm. of ACM*, 34(1), 1991, 38-58.
- [7] Ellis, C. A., and Wainer, J. A Conceptual Model of Groupware. *Proceedings of CSCW'94 (Chapel Hill NC, 1994)*, ACM Press, 79-88.
- [8] Fuks, H., Ryan M., and Sadler, M. Outline of a Commitment Logic for Legal Reasoning. *Proceedings of 3rd International Conference on Logics, Informatics and Law*, V2, (Florence, Italy, 1989), 391-405.
- [9] Fuks, H. Negotiation using Commitment and Dialogue. PhD. Thesis, Department of Computing, Imperial College, University of London. 1991.
- [10] Fuks, H., Gerosa, M. A., and Lucena, C. J. P. The Development and Application of Distance Learning on the Internet. *Open Learning - The Journal of Open and Distance Learning*, 17(1), 2002, 23-38.
- [11] Fuks, H., Raposo, A.B., and Gerosa, M.A. Engineering Groupware for E-Business. *Proceedings of EBR 2002 – First Seminar on Advanced Research in Electronic Business (Rio de Janeiro, Brazil, 2002)*, 78-84.
- [12] Koo, C., and Wiederhold, G. A commitment-based communication model for distributed office environments. *Proceedings of Office Information Systems, SIGOIS Bulletin*, 9(2-3), 1988, 291-298.
- [13] Laufer, C. C., and Fuks, H. ACCORD: Conversation Clichés for Cooperation. *Proceedings of COOP'95 – First International Workshop on the Design of Cooperative Systems (Antibes-Juan-les-Pins, France, 1995)*, INRIA Press, 351-369.
- [14] Malone, T. W., and Crowston, K. What is coordination theory and how can it help design cooperative work systems? *Proceedings of CSCW'90 (Los Angeles CA, 1990)*, ACM Press, 357-370.
- [15] Malone, T. W., and Crowston, K. The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26 (1), 1994, 87-119.
- [16] Raposo, A. B., and Fuks, H. Defining Task Interdependencies and Coordination Mechanisms for Collaborative Systems. In Blay-Fornarino, M. et al. (Eds.), *Cooperative Systems Design. Frontiers in Artificial Intelligence and Applications*, 74, 88-103. IOS Press, Holland. 2002.
- [17] Raposo, A.B., Magalhães, L.P., and Ricarte, I.L.M. Petri Nets Based Coordination Mechanisms for Multi-Workflow Environments. *International Journal of Computer Systems Science & Engineering* 15(5), 2000, 315-326.

- [18] Schmidt, K. Riding A Tiger, Or Computer Supported Cooperative Work. Proceedings of The 2nd European Conference on Computer-Supported Cooperative Work (Amsterdam, 1991). Kluwer Academic Pubs, 1-16.
- [19] Schmidt, K., and Bannon, L. J. Taking CSCW seriously - Supporting articulation work. Computer Supported Cooperative Work, 1(2), 1992, 7-40.
- [20] Schmidt, K., and Simone, C. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. Computer Supported Cooperative Work, 5(2-3), 1996, 155-200.
- [21] Schmidt, K., and Simone, C. Mind the gap! Towards a unified view of CSCW. Proceedings of COOP 2000 - 4th International Conference on the Design of Cooperative Systems (Sophia Antipolis, France, 2000), 205-221.
- [22] Simone, C., Mark, G., and Giubbilei, D. Interoperability as a Means of Articulation Work. Proceedings of WACC: Int. Conf. on Work Activities Coordination and Collaboration (1999), 39-48.
- [23] Suchman, L. A. Do Categories Have Politics? Computer Supported Cooperative Work, 2(3), 1994, 177-190.
- [24] Tapscoot, D., Ticoll, D., and Lowy, A. Digital Capital: Harnessing the Power of Business Webs. Harvard Business School Press, USA, 2000.
- [25] Winograd, T., and Flores, F.: Understanding Computers And Cognition. Addison-Wesley, USA, 1987.
- [26] Winograd, T. Categories, Disciplines, and Social Coordination. Computer Supported Cooperative Work, 2(3), 1994, 191-197.
- [27] Woo, C.C. Sact - A Tool for Automating Semi-Structured Organizational Communication. Proc. of Conference on Office Information Systems (Cambridge, MA, 1990), ACM Press, 89-98