

Evolução Dinâmica de um Curso a Distância Modelado por Workflow¹

Graziela Flores Kunde

Maria Aparecida Souto

José Palazzo M. de Oliveira

Resumo - O objetivo deste artigo é caracterizar as mudanças dinâmicas que podem ocorrer no esquema conceitual de um curso para apoiar um ambiente na Web para oferecer o ensino em ambiente distribuído por meio de navegadores. Esta atividade faz parte do Projeto Tapejara/ProTeM CNPq-CC. O conteúdo do curso é representado conceitualmente e executado como um workflow². São analisadas as modificações no plano do curso (esquema conceitual) e as repercussões destas modificações nas atividades dos alunos inscritos. A utilização da tecnologia de workflow com suporte à evolução de esquemas permite uma melhor interação entre o professor e os alunos e uma maior flexibilidade na construção e manutenção de cursos a distância.

Abstract - The main goal of this paper is the characterization of the dynamic evolution in the conceptual schema of a Web to support distributed learning through browsers. The course content is conceptually modelled and executed as a workflow. A detailed analysis is developed to characterise the course conceptual schema modifications and the repercussion in the students' activities. The use of workflow technology with evolution enables a best interaction between the teacher and the students, and a bigger flexibility on the construction and maintenance of distance courses.

1 INTRODUÇÃO

A World Wide Web (WWW) é hoje o instrumento com maior condição para promover o ensino a distância, através da disponibilização de aplicações educacionais interativas. Esta modalidade de ensino requer uma seqüência formal de atividades de modo a possibilitar não só a apresentação dos conteúdos, como também a avaliação dos esforços individuais dos estudantes e dos resultados obtidos. Por outro lado, é necessária uma ampla possibilidade de acompanhamento dos conteúdos para atender às características de aprendizado peculiares a cada aluno. A coordenação da execução de um curso passa a constituir uma importante característica a ser considerada no planejamento das atividades. Isto é consequência da liberdade do aluno em cursos na Web para navegar através dos conteúdos, gerando um grande problema, uma vez que o estudante pode ser tentado a “surfear” no mar de material disponível, ao invés de seguir as lições (Casati e Pernici 96).

¹ Este trabalho foi parcialmente financiado pelo Projeto ProTeM-CC do CNPq

² Por *workflow* entende-se o resultado da modelagem formal das atividades desenvolvidas em uma organização. Esta modelagem inclui as atividades administrativas estruturadas bem como as atividades de tomada de decisão. Associada à modelagem existe uma tecnologia de computação que dá o suporte para a execução do trabalho. O termo *workflow*, em inglês, está consagrado na área da Informática para representar os sistemas que tratam do *fluxo de trabalho*.

Neste contexto, as técnicas de modelagem de *workflow*, originadas e amplamente utilizadas para a modelagem de negócios e de processos, têm-se apresentado como uma possibilidade para modelar as atividades inerentes aos cursos de ensino a distância, uma vez que permitem a definição do processo de ensino como um todo, com a especificação clara de todas as atividades que devem ser executadas, de seus relacionamentos e os agentes responsáveis por sua execução. O *workflow* é uma alternativa de geração de formas alternativas de apresentação do conteúdo e de monitoração do curso, facilitando o acompanhamento dos alunos pelo professor, que possui uma visão global das atividades. Além disso, a manutenção do material torna-se mais fácil e o aluno fica devidamente orientado na seqüência de caminhos a seguir.

Alguns trabalhos que estendem os conceitos de *workflow* para a área de ensino a distância são referenciados em (Casati e Pernici 96), (Nicolao et al. 98) e (Oliveira et al. 98, Sizilio 00). Estes artigos apresentam, também, modelos de cursos através de *workflow*.

A tecnologia de *workflow* para modelar cursos a distância resolve o problema da coordenação das atividades. Porém, a rigidez imposta por um esquema de curso pré-definido (modelo de *workflow*) com as atividades didáticas previamente programadas pode limitar, em alguns aspectos, o rendimento do curso. Muitas vezes um curso possui longa duração e características dinâmicas, necessitando de modificações no conteúdo programático enquanto os alunos estão cursando. Isto é especialmente visível em áreas tecnológicas de rápida evolução como a Informática e a Bio-tecnologia.

Para tratar deste problema foi desenvolvida uma arquitetura, baseada na modelagem do curso por meio de *workflow* (Nicolao et al. 98) e de agentes inteligentes (Oliveira et al. 98), destinada a acompanhar os alunos, aliviando parcialmente os professores das *atividades de acompanhamento rotineiro* e suportando modificações dinâmicas no plano do curso. Esta tecnologia permite uma atenção constante aos alunos com a adaptação do "Plano de Curso" às características do conjunto de alunos atendidos. O professor, ao longo do curso, tem a possibilidade de realizar mudanças, como por exemplo, a inclusão de uma nova

atividade didática, direcionando suas aulas de acordo com as necessidades identificadas. Complementarmente, esta tecnologia permite levar ao professor informações detalhadas sobre cada aluno, possibilitando uma maior individualidade no atendimento, mesmo com o aumento muito significativo da população atendida em cada curso. Desta forma, aliam-se as características de atendimento individual com a necessidade atual de atender a um grande número de alunos, procurando diminuir o número de excluídos do processo tradicional de ensino que obriga a presença física em sala de aula e em horários pré-definidos.

Este trabalho visa, além de reafirmar a tecnologia de *workflow* como uma forma de coordenação dos processos de um curso, oferecer uma abordagem onde o processo pode ser modificado dinamicamente sempre que houver necessidade, sem afetar o andamento dos alunos.

O objetivo deste artigo é caracterizar as mudanças dinâmicas mais significativas que podem ocorrer no esquema conceitual de um curso com a análise da repercussão das mudanças nas instâncias em execução, que representam os alunos seguindo o curso. O estudo de caso apresentado enfoca a modelagem conceitual das atividades de ensino de um curso a distância através de técnicas de *workflow* evolutivo. Procurou-se apresentar o assunto sem uma excessiva ênfase nos aspectos tecnológicos da modelagem em atenção aos leitores com formação na área de Educação. Mas em temas multidisciplinares como o presente, é impossível eliminar completamente o viés da área principal do trabalho. Este trabalho está integrado no Projeto Tapejara, financiado pelo Programa ProTeM-CC do CNPq.

O artigo está organizado da seguinte forma: a seção 2 apresenta alguns conceitos relativos à modelagem de *workflow*. A seção 3 mostra o problema da propagação de mudanças dinâmicas, comentando alguns trabalhos encontrados na literatura que enfocam o assunto. A seção 4 mostra o modelo completo do curso. A análise de modificações que podem ocorrer no esquema conceitual do curso a distância é apresentada na seção 5. A seção 6 relata o processo de modificação de um *workflow* representando um curso a distância. Finalmente, a seção 7 apresenta as conclusões obtidas.

2 MODELAGEM DE WORKFLOW

Um *workflow* é um modelo computacional de um processo do mundo real e permite descrever, ao mesmo tempo, ações humanas, como as ações dos alunos no curso, e ações automáticas, como atualizações em bancos de dados, por exemplo. Esta descrição de curso possui um esquema conceitual composto por tarefas, que representam as ações. As tarefas são executadas em uma ordem especificada pela descrição do fluxo de trabalho (*workflow*) e são desenvolvidas, geralmente, por agentes diferentes, em locais possivelmente afastados um do outro. Conectores são utilizados para definir possíveis caminhos de roteamento entre as tarefas (Casati e Pernici 96).

Ao longo deste artigo será utilizado o modelo de Casati/Ceri (Casati et al. 95) para a representação dos esquemas conceituais relativos ao curso. Neste modelo, uma **tarefa** é uma unidade elementar de trabalho, representada por uma caixa. No curso, todas as atividades desempenhadas pelo aluno, pelo professor, ou exclusivamente pelo sistema, são consideradas tarefas. As seqüências e dependências entre as tarefas podem ser especificadas de várias formas: duas tarefas podem ser diretamente conectadas por um arco, com o significado que, tão logo a primeira termine, a segunda estará pronta para execução. Em todos os outros casos, conexões entre tarefas são realizadas por duas tarefas de roteamento de propósitos especiais: *forks*, para inicialização de execuções concorrentes (*splits* na definição WfMC (WMC 96)), e *joins*, para sincronização (junção) após a execução concorrente.

Outro conceito utilizado no trabalho, é o conceito de **supertarefa**. A supertarefa permite a definição de um módulo *workflow* com um esquema próprio e constitui-se de um conjunto de tarefas relacionadas, que pode ser considerado como uma unidade de trabalho em um nível mais alto de abstração, e ser atribuído a um agente responsável.

O modelo também possui o conceito de **papel**, um conjunto de atores (participantes) que apresentam atributos e características que os tornam aptos a executar a tarefa relacionada ao papel. Um mesmo participante pode executar mais de um papel. No estudo de caso três papéis são definidos: aluno, professor e sistema.

3 PROPAGAÇÃO DE MODIFICAÇÕES NO ESQUEMA E NAS INSTÂNCIAS ATIVAS

As modificações que ocorrem ao longo da apresentação do curso refletem-se em mudanças no esquema conceitual do *workflow*. Devido a estas mudanças, o sistema de *workflow* deve evoluir. Dois tipos de mudança são os principais fatores que geram a necessidade de evolução em sistemas de *workflow* (Han et al. 98). O primeiro é constituído pelas mudanças do ambiente, pois as atividades de ensino são altamente dinâmicas e sujeitas a evolução constante. No caso em estudo estas atividades correspondem à apresentação do conteúdo de um curso. O segundo tipo de mudanças que reforçam a necessidade de evolução de *workflow* são os avanços técnicos no ambiente de suporte. Os sistemas de software são confrontados com requisitos de evolução causados por avanços técnicos que freqüentemente conduzem à reconfiguração de sistemas devido a, por exemplo, substituição e atualização de componentes de software, adição de novos componentes e mudanças em interfaces componentes.

Segundo Casati et al (1996), o problema da evolução de *workflow* possui duas facetas:

- ◆ **Evolução estática de workflow** - refere-se à questão da modificação da descrição do *workflow*;
- ◆ **Evolução dinâmica de workflow** - refere-se ao gerenciamento de instâncias em execução de um *workflow* cuja descrição foi modificada. No contexto do curso, a evolução estática refere-se à modificação do plano de curso e a evolução dinâmica refere-se ao gerenciamento das modificações nos processos do curso em andamento, adaptação dos alunos em curso às modificações no plano de curso.

Uma das questões mais desafiadoras na modificação de *workflows* é o gerenciamento de execuções iniciadas com o modelo antigo. Para suportar a evolução dinâmica é necessário um profundo entendimento das repercussões que estas mudanças provocam nos esquemas e nas correspondentes instâncias em execução. Soluções simples, como deixar o processo terminar de acordo com o modelo antigo ou abortá-lo, são

geralmente inconvenientes. Além disso, em certos processos é impossível parar as atividades para organizar a mudança.

Duas abordagens para mudanças dinâmicas em *workflow* são descritas em Casati et al. (1996), e Reichert (1997). A primeira salienta a importância de um efetivo gerenciamento da evolução dinâmica. É proposto um conjunto de primitivas da linguagem WFML (*Workflow Modification Language*) que permitem modificações genéricas de *workflow*, preservando a correção sintática tanto na parte estática quanto na dinâmica. Também é introduzida uma taxonomia que descreve como as instâncias em execução podem ser manejadas após uma modificação e as restrições de integridade que estas instâncias devem obedecer para continuar a execução de acordo com o modelo modificado.

A segunda abordagem enfoca sistemas que suportam a especificação do *workflow* em tempo de execução. Também possui um conjunto de operações para mudanças estruturais dinâmicas, porém enfatiza mudanças específicas em instâncias em execução, além das mudanças no esquema do *workflow* e sua propagação nas instâncias. A maior contribuição desta abordagem para o presente trabalho é o conceito de interdependência entre as tarefas e as regras definidas com relação ao fluxo de dados. O resultado da interdependência é a manutenção da integridade após as modificações dos dados de entrada e saída das tarefas e das bases de dados. Algumas restrições estruturais em relação ao esquema do *workflow* também podem ser adaptadas para mudanças no esquema e instâncias.

Horn e Jablonski (1998) classificam as modificações nas instâncias em execução que iniciaram segundo um modelo que é posteriormente modificado em três classes: (i) a parte modificada do esquema afeta uma parte já executada da instância; (ii) a parte modificada do esquema afeta uma parte que está sendo executada momentaneamente e (iii) a parte modificada do esquema afeta uma parte da instância que deve ser executada no futuro. No primeiro caso, como a mudança dá-se num ponto já executado (o ponto da modificação é anterior ao ponto de execução), a adaptação da instância para a nova especificação não é possível, a menos que algumas atividades sejam desfeitas ou que sejam inseridas atividades compensatórias. No segundo caso, como a modificação afeta uma parte

que está em execução, duas situações podem ocorrer: o ponto de modificação inclui o ponto de execução, sendo necessário que parte da atividade seja repetida ou a execução da instância ainda não alcançou as partes modificadas do *workflow*, sendo a adaptação aplicável. No terceiro caso, é possível aplicar a adaptação na instância em execução, pois o ponto de execução é posterior à mudança. A princípio, esta análise serve para qualquer instância em execução. Porém, outros fatores devem ser levados em conta, como a dependência de dados entre tarefas, conforme Reichert (1998).

4 EVOLUÇÃO DE UM CURSO

Um estudo de caso foi elaborado para demonstrar as mudanças dinâmicas que podem ocorrer em um modelo de um curso a distância e a propagação destas mudanças nas instâncias em execução. Os alunos *podem iniciar as aulas a qualquer momento*, não havendo prazos definidos para a inscrição. Com isso, cada aluno no curso será representado por uma instância descrita pelo modelo e terá um andamento próprio. Este fato obriga a ser considerada a evolução dinâmica, pois há uma entrada contínua de alunos e, a cada alteração do curso, diferentes instâncias estarão em execução (alunos cursando) em diversos estágios de andamento. Para o gerenciamento de mudanças neste caso, todas as instâncias em execução deverão ser levadas em conta e as técnicas propostas deverão ser aplicadas.

4.1 MODELO DO CURSO

O modelo do curso é mostrado na figura 1. De uma maneira geral, o curso é representado pelo seguinte processo: em um primeiro momento o aluno deverá inscrever-se no curso. Para isto, deverá fornecer dados cadastrais, que serão armazenados (dados do aluno). Cada aluno receberá um número de matrícula único (*id_aluno*). Após a inscrição, o aluno deverá responder a um questionário para avaliar o seu nível de conhecimento. A avaliação do questionário, realizada pelo sistema, irá determinar se o aluno possui condições de iniciar a primeira aula do curso ou se deverá participar de aulas de nivelamento. Se o nivelamento for necessário (a tarefa *Avaliação* atribuiu 'insuficiente' à variável *nível*), o aluno deverá percorrer algumas atividades extras. Após este módulo de

nivelamento, o aluno irá participar de um conjunto de módulos de estudo, cada qual com um assunto e atividades próprias. A cada módulo de estudo, é registrado o desempenho do aluno, através de testes, participações e exercícios. Após o término de todas os módulos, será calculado o rendimento do aluno no curso, através da tarefa *Cálculo da média*, com base nos dados armazenados em registro de atividades. A emissão do certificado é a última tarefa do processo, que irá fornecer o conceito final calculado na tarefa anterior.

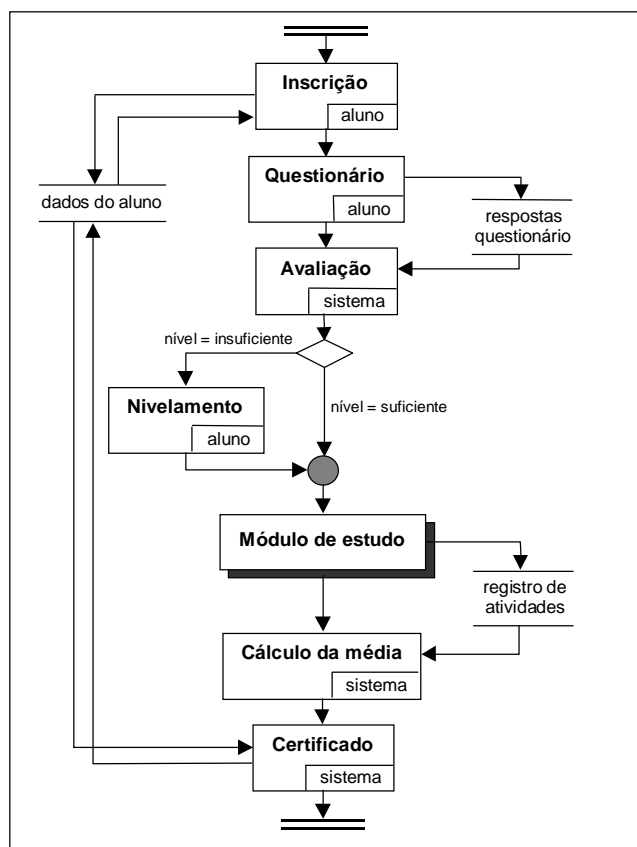


Figura 1 – Modelo do curso

O aluno não é obrigado a concluir um módulo de estudo no mesmo dia em que iniciou. Ele pode fracioná-lo da maneira que desejar. Tecnicamente, isto requer que a cada vez que este aluno conectar-se, deverá ser verificado seu número de matrícula e senha e, após, ser recuperado o estado de andamento de seu curso.

No esquema conceitual do curso, uma tarefa simples é representada por uma caixa contendo o nome da tarefa e o papel quem a desempenha, ou seja, o agente que executa a tarefa (aluno, professor e sistema). Os papéis são representados por uma pequena

caixa situada no canto inferior direito da tarefa contendo a indicação do papel. As caixas sombreadas representam supertarefas, que são abstrações de um conjunto de tarefas. Para as supertarefas os papéis não são indicados, pois normalmente são constituídas por diferentes tarefas desempenhadas por diferentes papéis. Os dados são representados por duas linhas horizontais, contendo o nome do dado em letras minúsculas. As setas ligando os dados às tarefas representam o fluxo de dados.

O esquema do curso basicamente possui informações pertinentes às atividades desempenhadas pelo aluno durante o curso. Também não foi representada pelo sistema a desistência do curso por parte do aluno. Este pode tomar tal decisão a qualquer momento. Portanto, há um caminho para o término do curso partindo de cada tarefa. Este caminho não foi representado para deixar o esquema mais claro.

Os módulos de estudo do curso possuem uma estrutura básica. O curso geralmente contém vários módulos, sendo o número definido no projeto do curso. Assim, o esquema do módulo de estudo representado pela supertarefa *Módulo de estudo* (figura 2) irá se repetir quantas vezes for o número total de módulos do curso, sendo cada ocorrência diferente apenas no conteúdo.

Um módulo de estudo é composto basicamente pela apresentação do assunto a ser tratado, um conjunto de *apontadores* que o aluno deverá seguir, um conjunto de textos relativos ao assunto, uma lista de discussão para participar, uma série de exercícios sobre o assunto e, finalmente, a avaliação do rendimento do aluno naquela aula, realizado pelo sistema.

Um módulo de estudo inicia com a apresentação do assunto. Esta apresentação é representada pela supertarefa *Apresentação do assunto*, figura 3. Primeiro, o aluno assistirá a uma apresentação em vídeo do professor, expondo o assunto de estudo e, a seguir, deverá ler um texto inicial obrigatório.

Após esta atividade, o aluno deverá acessar, na *Web*, pelo menos dois *apontadores* de um conjunto proposto. A figura 4 representa a supertarefa *Conjunto de apontadores*.

Após seguir os *apontadores*, o aluno deverá participar de atividades de leitura de uma seqüência de tex-

tos, figura 5. Paralelamente, o aluno deverá selecionar os tópicos importantes e descrevê-los. Estes tópicos ficarão armazenados em *tópicos* para utilização futura.

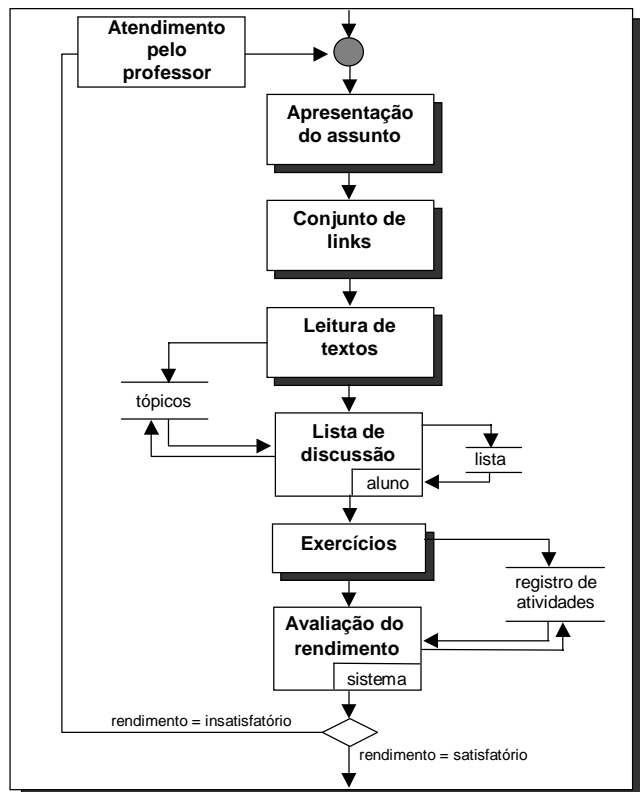


Figura 2 – Supertarefa *Módulo de estudo*

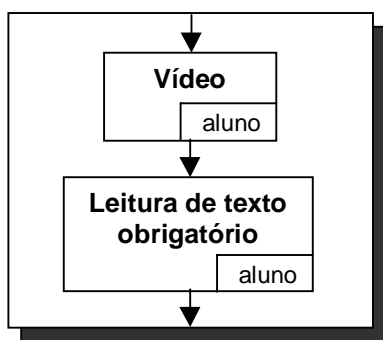


Figura 3 – Supertarefa *Apresentação do assunto*

Terminadas as atividades de leitura de textos e seleção de tópicos importantes, o aluno participará de uma lista de discussão (tarefa *Lista de discussão*), onde colocará a sua contribuição sobre o assunto, ressaltando os tópicos que selecionou e constatará o que os colegas também selecionaram.

Após participar da lista de discussão, o aluno deverá realizar alguns exercícios, através da supertarefa

Exercícios. Esta consiste inicialmente em duas tarefas que podem ser executadas em paralelo (*Exercícios objetivos* e *Leitura de texto*).

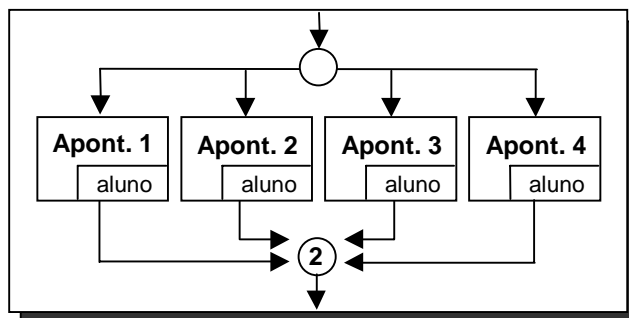


Figura 4 – Supertarefa *Conjunto de apontadores*

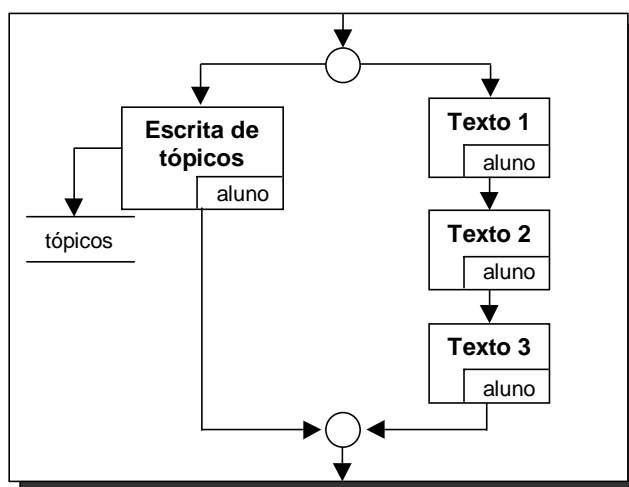


Figura 5 – Supertarefa *Leitura de textos*

Na tarefa *Avaliação*, o sistema automaticamente calcula o rendimento do aluno nos exercícios objetivos, atribuindo uma nota que pode variar entre 0 e 10. Se o aluno obtiver nota menor que 7, deverá realizar tarefas de recuperação (tarefa *Recuperação*) determinadas em função do perfil de aprendizado demonstrado pelo aluno e dos erros cometidos. Paralelamente, o aluno, após realizar a leitura, passará para uma série de exercícios de interpretação de texto (tarefa *Exercícios de interpretação*), que serão corrigidos futuramente pelo professor (tarefa *Correção*). Analogamente as outras tarefas, os exercícios realizados pelo aluno são armazenados em *Registro de atividades*. Estando prontas as tarefas *Avaliação*, *Recuperação* (caso esta tarefa tenha sido habilitada por um desempenho insuficiente nos *Exercícios objetivos*) e *Correção*, o aluno poderá consultar os exercícios corrigidos (tarefa *Exercícios corrigidos*). Este subprocesso é representado na figura 6.

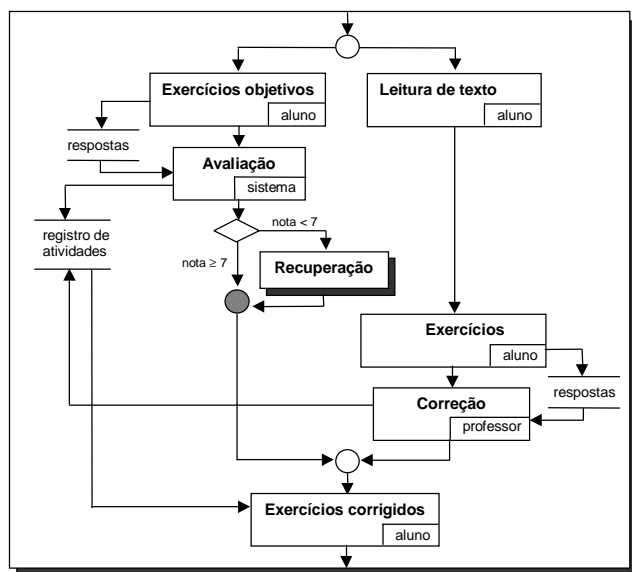


Figura 6 – Supertarefa *Exercícios*

Uma observação deve ser feita com relação à tarefa *Correção*. Por ser uma tarefa executada pelo professor, são necessárias restrições temporais ou períodos de espera por parte do aluno, pois o professor não estará necessariamente conectado no momento em que o aluno terminar os exercícios de interpretação. Estas restrições temporais não foram representadas, pois estão fora do escopo deste trabalho.

Terminados os exercícios, o sistema, com base em *registro de atividades*, calcula o rendimento do aluno naquele módulo de estudo, armazenando-o também. Se o rendimento não for satisfatório, o sistema atribui o conceito *‘insatisfatório’* à variável condição *rendimento* e o aluno deverá repetir o mesmo módulo de estudo. Esta situação é representada através da *realimentação* após a avaliação da condição, passando por um atendimento personalizado pelo professor, figura 2. Caso contrário, a variável recebe o valor *‘satisfatório’*, estando o aluno apto a cursar o próximo módulo de estudo do curso.

5 MUDANÇAS NO ESQUEMA DE UM CURSO A DISTÂNCIA

São várias as modificações que podem ocorrer no esquema conceitual de um curso a distância representado por um *workflow*. Nesta seção são demonstradas as modificações mais significativas. Um estudo da completude e minimalidade destas modificações necessita de um formalismo rigoroso e, por tratar-se

de Ciência da Computação, está fora do escopo desta revista.

5.1 INSERÇÃO DE TAREFA SEQUENCIAL

Para exemplificar a inserção de uma tarefa sequencial, pode-se considerar, por exemplo, a tarefa *Leitura de textos*. O professor pode chegar à conclusão que os alunos deverão ler um novo texto, após o segundo texto e antes do terceiro. O novo esquema para esta modificação (adição da tarefa *Texto* como sucessora de *Texto 2* e predecessora de *Texto 3*) é representado pela figura 7.

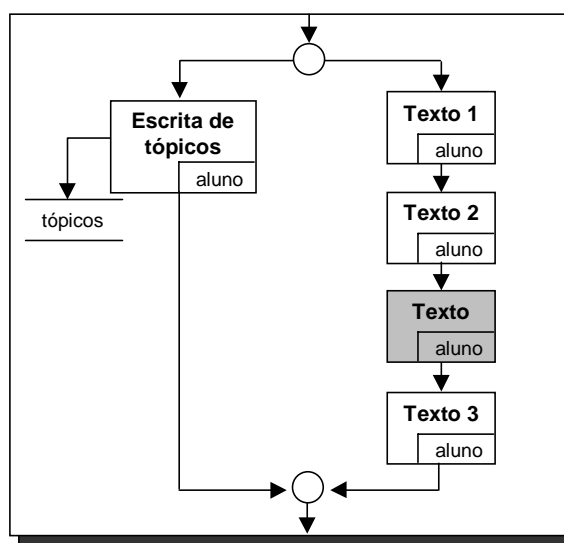


Figura 7 – Inserção da tarefa *Texto*

a) Restrições

Para esta modificação de esquema, as seguintes restrições podem ser definidas:

- ◆ As tarefas *Texto 2* (predecessora da nova tarefa) e *Texto 3* (sucessora) deverão existir;
- ◆ A tarefa *Texto 3* não poderá preceder a tarefa *Texto 2*, pois ciclos ruins poderiam levar a bloqueios de execução.

b) Instâncias em execução

No momento da modificação do esquema, existem vários alunos (instâncias) em diferentes níveis no curso. Portanto, três grupos podem ser observados:

- ◆ **Grupo 1** - As instâncias nas quais a execução ocorrer em *Texto 1* ou *Texto 2* (ou em tarefas anteriores) poderão refletir a mudança, com a habi-

litação da nova tarefa logo após o término de *Texto 2*;

- ◆ **Grupo 2** - As instâncias nas quais a tarefa *Texto 3* estiver em execução não poderão sofrer propagação direta da modificação;
- ◆ **Grupo 3** - As instâncias nas quais a execução ocorrer em um ponto posterior à tarefa *Leitura de textos* não poderão sofrer a propagação direta da modificação.

Dentre as soluções estudadas para o gerenciamento das instâncias, pode-se analisá-las de acordo com a semântica do processo:

- ◆ **Abortar o processo** - inviável neste caso, pois significaria terminar o curso para os alunos pertencentes a este grupo. Haveria uma perda enorme de trabalho já realizado e uma grande insatisfação por parte dos alunos;
- ◆ **Deixar a instância continuar a execução seguindo o modelo antigo** - em algumas situações esta medida pode ser considerada. Talvez tal solução seja aplicável ao Grupo 3, cuja inserção da nova tarefa acarretaria um trabalho maior, muitas vezes sendo necessário voltar atrás de um estágio muito avançado;
- ◆ **Voltar alguns passos** - para o Grupo 2 esta medida é aplicável, pois voltar atrás não acarretará muitos problemas, uma vez que será necessário retroceder apenas na leitura de um texto. Para o Grupo 3, porém, pode não ser vantajoso voltar muitos passos, atrasando o andamento do aluno no curso;
- ◆ **Criar um esquema temporário para compensar a modificação** - esta solução parece ser a mais apropriada para o Grupo 2 se a ordem de leitura dos textos não for essencial. Portanto, um esquema alternativo seria o de inserir a nova tarefa após a tarefa *Texto 3*. Para o Grupo 3, a inserção de uma tarefa de leitura de texto deveria, se possível, ser feita paralelamente a outra tarefa.

c) Fluxo de dados

A inserção da tarefa sequencial *Texto* não acarreta problemas no fluxo de dados do curso.

5.2 INSERÇÃO DE TAREFA PARALELA OPCIONAL

No caso de uma inserção de tarefa paralela a outras pode-se analisar, como exemplo, a supertarefa *Conjunto de apontadores*. O professor pode desejar indicar mais um *apontador* além dos existentes como nova opção. A adição da tarefa *apontador* como sucessora de *Leitura de texto obrigatório* e predecessora de *Escrita de tópicos* e *Texto 1* é apresentada na figura 8.

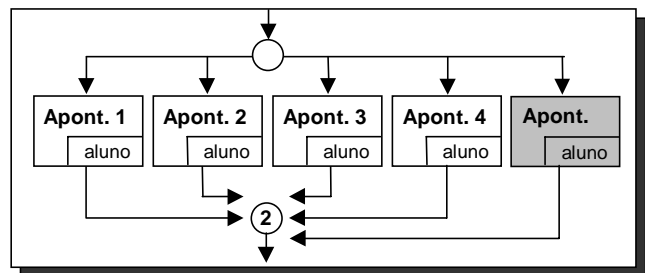


Figura 8 – Inserção de um *apontador* opcional

a) Restrições

Para esta modificação de esquema, as seguintes restrições podem ser definidas:

- ◆ A tarefa predecessora *Leitura de texto obrigatório* e as tarefas sucessoras *Escrita de tópicos* e *Texto 1* deverão existir;
- ◆ As tarefas *Escrita de Tópicos* e *Texto 1* não poderão preceder a tarefa *Leitura de texto obrigatório*;
- ◆ As tarefas *Escrita de tópicos* e *Texto 1* não poderão ser sucessoras ou predecessoras uma da outra.

b) Consistência comportamental/instâncias em execução

Esta inserção não provoca inconsistências caso as instâncias já tenham passado pela supertarefa, pois o aluno pode optar pelos outros *apontadores* já recomendados. Porém, para aqueles alunos que desejarem seguir o novo *apontador* e já passaram pelas tarefas, passos deverão ser desfeitos ou um esquema temporário alternativo com a sugestão do novo *apontador* deverá ser introduzido para compensar a modificação.

c) Fluxo de dados

A inserção de um novo *apontador* opcional não acarreta problemas no fluxo de dados do curso.

5.3 INSERÇÃO DE TAREFA PARALELA OBRIGATÓRIA

A figura 9 mostra o caso de o novo apontador inserido ser obrigatório. Uma estrutura de junção (*join*) (Casati et al. 95) foi inserida para que as próximas tarefas (da supertarefa *Leitura de textos*) somente sejam disparadas com o término de dois apontadores (do conjunto) e da nova tarefa (novo apontador). Portanto, além do aluno optar por dois dos quatro apontadores do modelo antigo, obrigatoriamente deve seguir o novo apontador.

a) Restrições

Para esta modificação de esquema, as seguintes restrições podem ser definidas:

- ◆ A tarefa predecessora *Leitura de texto obrigatório* e as tarefas sucessoras *Escrita de tópicos* e *Texto 1* deverão existir;
- ◆ As tarefas *Escrita de Tópicos* e *Texto 1* não poderão preceder a tarefa *Leitura de texto obrigatório*;
- ◆ As tarefas *Escrita de tópicos* e *Texto 1* não poderão ser sucessoras ou predecessoras uma da outra;
- ◆ As tarefas sincronizadas pela nova junção (*join*) deverão existir e não poderão ser sucessoras das tarefas *Escrita de tópicos* e *Texto 1*.

b) Instâncias em execução

Quanto às instâncias em execução, dois grupos podem ser observados:

- ◆ **Grupo 1** - As instâncias nas quais a execução ocorrer nos *apontadores* do conjunto ou em tarefas anteriores, poderão refletir automaticamente a mudança, com a execução da nova tarefa logo após dois *apontadores* terminados;
- ◆ **Grupo 2** - As instâncias nas quais a execução ocorrer em uma tarefa subsequente não poderão sofrer a propagação direta da modificação.

Dentre as soluções estudadas para o gerenciamento das instâncias, pode-se analisá-las, de acordo com a semântica do processo:

- ◆ **Abortar o processo** - inviável neste caso, pois significaria terminar o curso para os alunos pertencentes a este grupo;
- ◆ **Deixar a instância continuar a execução seguindo o modelo antigo** - Assim como na inserção de tarefa sequencial, talvez esta medida seja aplicável para o Grupo 2, cuja inserção da nova tarefa acarretaria um trabalho maior, muitas vezes sendo necessário voltar atrás de um estágio muito avançado;
- ◆ **Voltar alguns passos** - se não for necessário voltar muitos passos, esta medida pode ser aplicável para o Grupo 2;
- ◆ **Criar um esquema temporário para compensar a modificação** - esta solução parece ser a mais apropriada para o Grupo 2, pois com a inserção da tarefa paralela a outra tarefa, não há a necessidade de retroceder nas tarefas do curso.

c) Fluxo de dados

A inserção de um *apontador* obrigatório não acarreta problemas no fluxo de dados do curso.

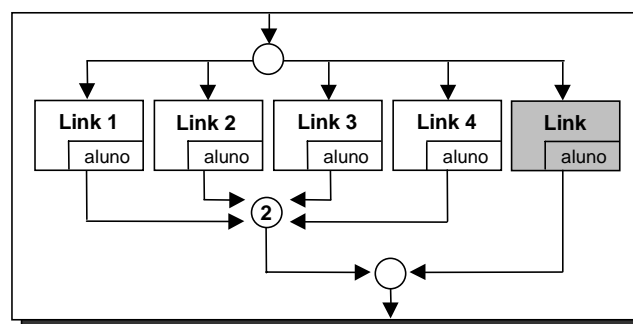


Figura 9 – Inserção de um *apontador* obrigatório

5.4 INSERÇÃO DE SUPERTAREFA CONDICIONAL

Para exemplificar esta mudança, pode-se supor que alguns alunos, após o módulo de exercícios, queiram enviar algumas dúvidas para serem respondidas por um professor. A modificação do esquema, portanto, corresponde à inserção de uma supertarefa *Dúvidas*, como sucessora da tarefa *Exercícios* e predecessora

da tarefa *Avaliação do rendimento*, figura 10. Cabe ressaltar que as modificações envolvendo supertarefas comportam-se da mesma forma que modificações envolvendo tarefas simples. Portanto, são aplicáveis as mesmas regras.

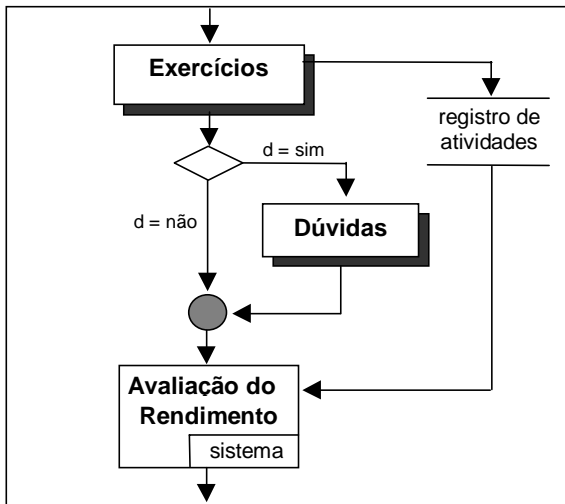


Figura 10 – Inserção da supertarefa *Dúvidas*

De acordo com a nova situação, se o aluno tiver dúvidas, a variável *d* recebe 'sim' e, após a conclusão dos exercícios, a condição é verificada e a supertarefa *Dúvidas* é ativada. Após o término desta nova supertarefa, a execução passa para a tarefa *Cálculo do rendimento*. Se o aluno não possui dúvidas, a variável recebe o valor 'não' e as tarefas da tarefa *Dúvidas* não são habilitadas.

a) Restrições

Para esta modificação de esquema, as seguintes restrições podem ser definidas:

- ◆ A tarefa predecessora *Exercícios* e a tarefa sucessora *Cálculo do rendimento* deverão existir;
- ◆ A tarefa *Cálculo do rendimento* não poderá preceder alguma subtarefa da tarefa *Exercícios*;
- ◆ Uma condição associada à nova tarefa deverá ser inserida.

b) Instâncias em execução

As instâncias em execução podem ser divididas em três grupos:

- ◆ **Grupo 1** - As instâncias nas quais a execução ocorrer na tarefa *Exercícios* ou anterior a ela, po-

derão refletir a mudança, com a avaliação da condição *d* após seu término;

- ◆ **Grupo 2** - As instâncias nas quais o estágio de execução for posterior à mudança, e para as quais o valor da variável *d* seria 'não' (o aluno já praticou os exercícios e não possui dúvidas), poderão continuar executando segundo o modelo antigo, pois o fluxo é o mesmo (após a tarefa *Exercícios* a tarefa *Cálculo do rendimento* é disparada);
- ◆ **Grupo 3** - As instâncias nas quais o estágio de execução for posterior à mudança, e para as quais o valor da variável *d* seria 'sim' (os alunos possuem dúvidas), não poderão adaptar diretamente a mudança.

Dentre as soluções estudadas para o gerenciamento das instâncias, pode-se analisá-las, de acordo com a semântica do processo:

- ◆ **Abortar o processo** - assim como para as outras modificações apresentadas, esta solução é inviável, pois significaria terminar o curso para os alunos pertencentes a este grupo;
- ◆ **Deixar a instância continuar a execução segundo o modelo antigo** - aplicável ao Grupo 2;
- ◆ **Voltar alguns passos** - se não for necessário voltar muitos passos, talvez esta medida seja aplicável ao Grupo 3;
- ◆ **Criar um esquema temporário para compensar a modificação** - esta solução parece ser aplicável ao Grupo 3, mas é mais apropriada se a tarefa *Cálculo do rendimento* não estiver finalizada. Caso contrário, a execução do processo já estará em alguma aula posterior.

c) Fluxo de dados

A inserção da supertarefa *Dúvidas* não acarreta problemas no fluxo de dados do curso.

5.5 REMOÇÃO DE TAREFA

Para exemplificar a remoção de uma tarefa, pode-se considerar, por exemplo, a supertarefa *Leitura de textos*. O professor decide que não é mais necessária a tarefa *Escrita de tópicos*. Portanto, os alunos devem

realizar somente a leitura dos textos indicados. O esquema resultante é apresentado na figura 11.

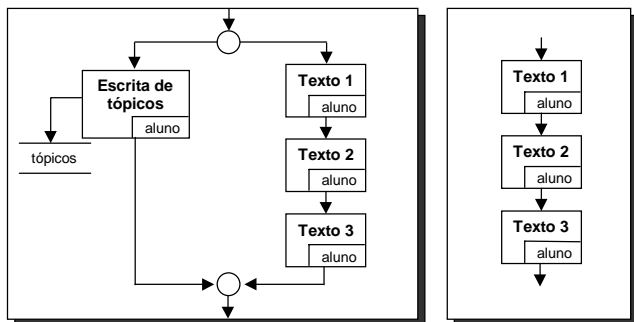


Figura 11 – Remoção da tarefa *Escrita de tópicos*

Neste caso, a tarefa retirada é executada concorrentemente ao conjunto de textos. Nota-se que esta mudança também provoca a retirada da estrutura chamada *fork total*, que disparava a execução concorrente, e do *join total* (Casati et al. 95), que sincronizava as execuções.

a) Restrições

Não há restrições estruturais genéricas que podem ser definidas para esta modificação.

b) Instâncias em execução

As instâncias em execução podem ser divididas em dois grupos:

- ◆ **Grupo 1** - As instâncias nas quais a execução ocorrer na tarefa Conjunto de apontadores ou anterior a ela poderão refletir a mudança, não ocorrendo a ativação de Escrita de tópicos;
- ◆ **Grupo 2** - As instâncias nas quais o estágio de execução for posterior à mudança não poderão adaptar diretamente a mudança.

Dentre as soluções estudadas para o gerenciamento das instâncias, pode-se analisá-las, de acordo com a semântica do processo:

- ◆ **Abortar o processo** - inviável neste caso, pois significaria terminar o curso para os alunos pertencentes a este grupo;
- ◆ **Deixar a instância continuar a execução segundo o modelo antigo** - é possível de ser aplicada ao Grupo 2, se for considerado que a execução desta tarefa não venha a prejudicar o curso;

- ◆ **Voltar alguns passos** - não é muito indicado, pois voltar atrás significa apenas repetir as outras tarefas sem a tarefa que se deseja retirar;
- ◆ **Criar um esquema temporário para compensar a modificação** - esta solução não é possível em se tratando da retirada de uma tarefa. Um esquema alternativo sem esta tarefa iria repetir as outras tarefas da tarefa *Leitura de textos*, sendo totalmente inviável.

c) Fluxo de dados

A remoção da tarefa *Escrita de tópicos* acarreta problemas no fluxo de dados, pois a tarefa *Lista de discussão* é considerada dependente de dados da tarefa a ser removida. Esta tarefa utiliza dados (tópicos) gerados por *Escrita de tópicos* e será executada com dados de entrada errados ou perdidos, caso não sejam tomadas medidas para tratar deste problema.

A solução neste caso, é a modificação de alguma tarefa para gerar estes dados, a modificação da tarefa *Lista de discussão* para que não utilize tais dados, ou a própria retirada desta tarefa dependente de dados. Para esta última alternativa, entretanto, deve-se levar em consideração todos os aspectos que envolvem a retirada de outra tarefa do esquema.

5.6 REMOÇÃO DE TAREFA OPCIONAL

Outro exemplo significativo é a remoção de *apontadores* do conjunto de *apontadores* propostos para a navegação. O professor decide, por exemplo, retirar o quarto *apontador* do conjunto. O esquema resultante desta modificação é representado pela figura 12.

Neste caso, a tarefa retirada é executada em paralelo aos outros *apontadores* do conjunto, porém a junção de tais tarefas é parcial, sendo necessário que apenas duas delas sejam terminadas.

a) Restrições

Para esta modificação de esquema, a seguinte restrição podem ser definida:

- ◆ A remoção não deverá ser permitida se a constante indicada no *join* parcial (Casati et al. 95) for igual ao número de tarefas do ramo paralelo, pois resultará em um número maior que o número de tarefas, provocando um erro de execução. Neste

caso, não há problemas, pois o valor desta constante é dois.

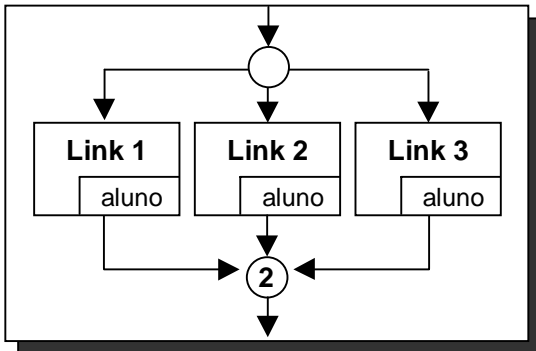


Figura 12 – Remoção de um apontador

b) Instâncias em execução

As instâncias em execução podem ser divididas em quatro grupos:

- ◆ **Grupo 1** - As instâncias nas quais a execução ocorrer na tarefa *Apresentação do assunto* ou anterior a ela, poderão refletir a mudança, não ocorrendo a ativação de *Apontador 4* após seu término;
- ◆ **Grupo 2** - As instâncias nas quais a execução ocorrer em *Conjunto de apontadores* não poderão refletir diretamente a mudança, pois *Apontador 4* já foi ativado. Porém, para a maioria dos casos, o fato da tarefa estar em execução não provoca problemas, bastando não ser terminada para que possa acompanhar a nova situação;
- ◆ **Grupo 3** - As instâncias nas quais o estágio de execução for posterior à mudança, e para as quais a tarefa *Apontador 4* não tenha sido terminada (situação causada pelo *join* parcial (Casati et al. 95)), poderão continuar sua execução sem necessidade de reparos;
- ◆ **Grupo 4** - As instâncias nas quais o estágio de execução for posterior à mudança, e para as quais a tarefa *Apontador 4* tenha sido executada, não poderão refletir fielmente a mudança.

Dentre as soluções estudadas para o gerenciamento das instâncias, pode-se analisá-las, de acordo com a semântica do processo:

- ◆ **Abortar o processo** - da mesma forma que para as outras modificações do curso é inviável, pois

significaria terminar o curso para os alunos pertencentes a este grupo;

- ◆ **Deixar a instância continuar a execução segundo o modelo antigo** - é possível de ser aplicada aos Grupos 2 e 3, pois para estas instâncias a tarefa em questão não foi terminada. Se a execução da tarefa *Apontador 4* não for prejudicial, esta solução também pode ser aplicável ao Grupo 4;
- ◆ **Voltar alguns passos** - não é muito indicado, pois voltar atrás significa apenas repetir as outras tarefas sem a tarefa que se deseja retirar;
- ◆ **Criar um esquema temporário para compensar a modificação** - esta solução não é indicada em se tratando da retirada de uma tarefa. Um esquema alternativo sem esta tarefa iria repetir as outras tarefas da tarefa *Conjunto de apontadores*, sendo totalmente desnecessário.

c) Fluxo de dados

A remoção da tarefa *Apontador 4* não acarreta problemas no fluxo de dados do curso.

5.7 DEPENDÊNCIA DE DADOS ENTRE AS TAREFAS DO CURSO

A inserção de tarefas também pode acarretar problemas relativos à dependência de dados entre as tarefas do curso. Um exemplo seria novamente a inserção das tarefas *Escrita de tópicos* e *Lista de discussão*, Figura 13. Se a execução da instância ocorrer na tarefa *Texto 3*, a inserção de *Escrita de tópicos* não será possível, mas a tarefa *Lista de discussão* seria teoricamente possível. Porém, como a tarefa *Escrita de tópicos* gera dados que são utilizados por *Lista de discussão*, esta tarefa seria executada sem os dados de entrada.

Uma solução para este problema seria voltar alguns passos, desfazendo as tarefas de textos, para que as duas novas atividades fossem inseridas no mesmo instante. Ou seja, para este caso, em relação ao fluxo de dados, não deverá ser permitida a inserção de parte da modificação proposta.

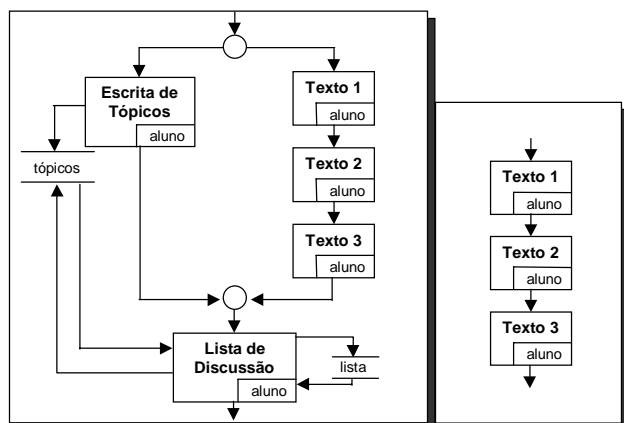


Figura 13 – Inserção das tarefas *Escrita de tópicos* e *Lista de discussão*

5.8 INSERÇÃO E REMOÇÃO DE SINCRONIZAÇÃO DO TIPO *FRACA*

Esta modificação refere-se ao caso particular de dois ramos de um processamento serem sincronizados através de uma estrutura de sincronização *fraca* (*soft*), representada por uma conexão pontilhada (Reichert et al. 98). A sincronização *fraca* representa uma dependência retardada entre duas tarefas n_1 e n_2 , isto é, n_2 pode ser executada se a tarefa n_1 é completada ou se não pode ser disparada mais. Portanto, este tipo de sincronização não requer necessariamente a execução com sucesso de n_1 . Esta modificação é realizada na supertarefa *Exercícios*, figura 14, e consiste em inserir um ramo *soft_sync* entre a tarefa *Recuperação* e a tarefa *Exercícios de interpretação*.

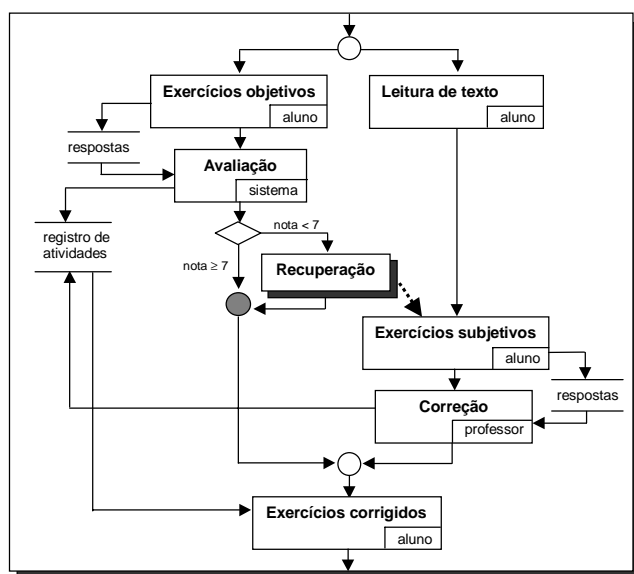


Figura 14 – Sincronização *fraca* em *Exercícios*

Com a inserção desta estrutura, se a tarefa *Recuperação* for habilitada, deve ser completada para a habilitação de *Exercícios de interpretação*. Com isto, o aluno, caso tenha obtido nota inferior a 7, deverá obrigatoriamente terminar a fase de recuperação para somente então iniciar os exercícios de interpretação. Inicialmente, como não havia tal estrutura, os exercícios de interpretação poderiam ser iniciados assim que o aluno terminasse a leitura do texto.

a) Restrições

Para esta modificação de esquema, as seguintes restrições podem ser definidas:

- ◆ A tarefa *Exercícios de interpretação* e a tarefa *Recuperação* deverão existir;
- ◆ A tarefa *Exercícios de interpretação* não poderá ser predecessora de alguma tarefa da tarefa *Recuperação*.

b) Instâncias em execução

As instâncias em execução podem ser divididas em nove grupos:

- ◆ **Grupo 1** - As instâncias nas quais a execução ocorrer na tarefa *Leitura de Texto* ou anterior a ela poderão refletir a mudança;
- ◆ **Grupo 2** - As instâncias nas quais a execução ocorrer na tarefa *Exercícios de interpretação* e para as quais a tarefa *Avaliação* não tenha sido concluída não poderão refletir automaticamente a mudança;
- ◆ **Grupo 3** - As instâncias nas quais a execução ocorrer na tarefa *Exercícios de interpretação* e para as quais a tarefa *Avaliação* tenha sido concluída e *Recuperação* estiver em execução ou já tenha sido concluída não poderão refletir a mudança sem mecanismos de reparo;
- ◆ **Grupo 4** - As instâncias nas quais a execução ocorrer na tarefa *Exercícios de interpretação* e para as quais a tarefa *Avaliação* tenha sido concluída e *Recuperação* não tenha sido habilitada poderão continuar executando de acordo com o modelo antigo;
- ◆ **Grupo 5** - As instâncias nas quais a execução ocorrer na tarefa *Correção* e para as quais a tarefa

Avaliação não tenha sido concluída não poderão refletir automaticamente a mudança;

- ◆ **Grupo 6** - As instâncias nas quais a execução ocorrer na tarefa *Correção* ou na tarefa *Exercícios Corrigidos* e para as quais a tarefa *Avaliação* tenha sido concluída e *Recuperação* estiver em execução ou já tenha sido concluída não poderão refletir a mudança sem mecanismos de reparo;
- ◆ **Grupo 7** - As instâncias nas quais a execução ocorrer na tarefa *Correção* ou na tarefa *Exercícios Corrigidos* e para as quais a tarefa *Avaliação* tenha sido concluída e *Recuperação* não tenha sido habilitada poderão continuar executando de acordo com o modelo antigo;
- ◆ **Grupo 8** - As instâncias nas quais a execução ocorrer em um ponto posterior à tarefa *Exercícios* e para as quais a tarefa *Recuperação* tenha sido executada e finalizada antes do início de *Exercícios de interpretação* poderão continuar executando de acordo com o modelo antigo;
- ◆ **Grupo 9** - As instâncias nas quais a execução ocorrer em um ponto posterior à tarefa *Exercícios* e para as quais a tarefa *Recuperação* tenha sido executada durante ou após *Exercícios de interpretação* não poderão adaptar a mudança.

Dentre as soluções estudadas para o gerenciamento das instâncias, pode-se analisá-las, de acordo com a lógica do processo:

- ◆ **Abortar o processo** - inviável neste caso, pois significaria terminar o curso para os alunos pertencentes a este grupo;
- ◆ **Deixar a instância continuar a execução segundo o modelo antigo** - indicado aos Grupos 4, 7, e 8;
- ◆ **Voltar alguns passos** - é indicado para os Grupos 3 e 6, uma vez que não é necessário desfazer grande parte de trabalho (para o Grupo 3 seria necessário desfazer parte de uma tarefa e para o Grupo 6 uma tarefa inteira e parte de outra);
- ◆ **Espera** - para os Grupos 2 e 5 inicialmente não é necessário desfazer passos. Estas instâncias deverão ser manejadas de modo que haja uma espera na execução dos exercícios de interpretação

(Grupo 2) e na tarefa *Correção* (Grupo 5), para a verificação da nota objetiva. Caso esta nota seja inferior a sete, passos deverão ser desfeitos para a conclusão da recuperação;

- ◆ **Criar um esquema temporário para compensar a modificação** - esta solução é indicada para o Grupo 9, pois estas instâncias estão em um estágio avançado de execução para as quais voltar atrás provocaria grande perda de trabalho já realizado. Um esquema alternativo para este grupo poderia conter outros exercícios de interpretação de reforço.

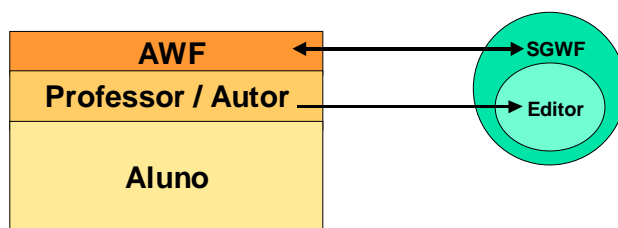
c) Fluxo de dados

A inserção de uma conexão *fraca* não acarreta problemas no fluxo de dados do curso.

6 PROCESSO DE MUDANÇA NO ESQUEMA DE UM CURSO A DISTÂNCIA

Em sistemas de *workflow* que suportam evolução, é importante a definição de quem realiza as mudanças dinâmicas, e como se dará o processo de mudança.

No estudo de caso apresentado neste artigo, observa-se três diferentes usuários no processo: o aluno, o professor e o AWF (administrador do *workflow*), como mostra a figura 15. Neste contexto, pode-se definir quem e como são realizadas as mudanças de esquema.



AWF – administrador do workflow
SGWF – sistema de gerência de workflow

Figura 15 – Níveis de usuários

Em um primeiro nível, o aluno é o usuário que utiliza o sistema, executando as atividades de estudo disponibilizadas pelo mesmo. Num segundo nível há o professor, que é autor/usuário porque, além de executar as atividades como orientador de ensino, possui meios de projetar a seqüência de atividades didáticas

e modificá-las sempre que houver necessidade. Os esquemas do curso podem ser realizados, por exemplo, através de um editor gráfico.

A composição do esquema deve ser suportada por uma metodologia, para que o professor faça a composição de descrições de tarefas. O plano resultante, então, deve ser mapeado para um modelo WF operacional.

Num terceiro nível há o AWF, usuário autorizado que possui conhecimentos mais técnicos sobre o processo e, no momento de uma mudança proposta pelo professor, segmenta as diversas instâncias em execução, ou seja, separa os alunos em grupos de acordo com o andamento de cada um, optando por políticas para cada grupo como, por exemplo, desfazer alguns passos (o que deve ser avisado ao aluno e ao professor) ou construindo esquemas alternativos para que compensem a modificação. O AWF é auxiliado pelo sistema, que fornece informações importantes para o tratamento das mudanças.

Portanto, o processo de mudança em um *workflow*, considerando o estudo de caso proposto, dá-se da seguinte forma: o professor observa a necessidade de uma mudança e altera o esquema do curso através de um editor alto nível. Automaticamente o SGWF (Sistema de Gerência de *Workflow*) verifica a consistência sintática e estrutural da modificação e as interdependências entre as tarefas ou supertarefas do processo, considerando o fluxo de dados. Auxiliado pelo SGWF, que também separa as instâncias em grupos de acordo com seus estágios de execução, o AWF verifica os grupos que podem propagar automaticamente as mudanças e os grupos que necessitam sua intervenção. Para estes, de acordo com as políticas para o gerenciamento de instâncias em execução Casati et al. (1996), (1998) determina as soluções, com a aprovação do professor, para a adaptação das mudanças.

7 CONCLUSÕES

Como o problema tratado é complexo e rico em estruturas no modelo de *workflow*, ele representou muitos aspectos dinâmicos que podem ocorrer nos sistemas. Este trabalho apresenta uma contribuição significativa no estudo e na apresentação de soluções para o problema da evolução de um esquema con-

ceitual de *workflow* quando existem instâncias em execução.

A análise realizada no estudo de caso mostrou, principalmente, as diferenças semânticas existentes em cada modificação de um processo de uma aplicação real. Com isso, confirma-se a importância de um usuário autorizado no processo de adaptação das instâncias em execução. Através desta análise, conclui-se que a solução apropriada para cada grupo de instâncias, que necessitam de intervenção do usuário autorizado, depende de quatro fatores:

- ◆ **Tipo e extensão das modificações** - em muitas situações a modificação possui uma abrangência muito pequena, comprometendo uma pequena parte da instância, sendo aceitável a decisão de deixar que a instância continue a execução segundo o modelo antigo. Em outras situações, a modificação afeta uma grande extensão da instância, sendo indispensável à adaptação da modificação, mesmo que isto provoque perda de trabalho devido a passos desfeitos ou compensados por outras atividades;
- ◆ **Motivação que provoca a mudança** - muitas vezes o motivo da mudança é um pequeno melhoramento, que pode ser descartado em instâncias cuja execução ocorre muito posteriormente ao ponto onde ocorreu a modificação. Em outras situações, a motivação da mudança é relacionada a um conceito novo a ser apresentado o que torna a adaptação da modificação imprescindível. Esta situação é corriqueira em áreas de avanço tecnológico rápido como a Informática, a Genética (Projeto Genoma, por exemplo) ou Tecnologia de Materiais;
- ◆ **História de execução da instância** - as instâncias podem estar em estágios de execução os mais diversos. Em situações cuja execução ocorre no final de um processo, por exemplo, pode-se considerar que não é vantajoso adaptá-lo à modificação.

Para que um sistema de *workflow* possua flexibilidade para acompanhar as mudanças do mundo real, e em particular, possa representar e gerenciar um curso a distância com entrada contínua de alunos, é necessário que possua meios para permitir que as alterações no esquema possam ser propagadas para as instâncias

de forma transparente e rápida. Porém, esta propagação não é trivial, pois pode causar erros ou inconsistências na execução. Muitos fatores devem ser levados em conta, como o estágio de execução de cada instância envolvida, a dependência de dados entre as diferentes atividades do processo, as formas de sincronização etc. É importante o profundo entendimento das mudanças dinâmicas que podem ocorrer em um sistema de *workflow* e as repercussões destas mudanças nas instâncias ativas. Por este motivo, foi construído um modelo de evolução de esquemas conceituais de *workflow*, descrito na seção 5, complexo e rico em estruturas para relacionar os principais tipos de mudanças passíveis de ocorrerem no esquema conceitual e, ao mesmo tempo, tratar as repercussões destas mudanças nas instâncias ativas do modelo.

A abordagem proposta alia os benefícios da modelagem de *workflow* para cursos a distância com o suporte a mudanças dinâmicas, necessidade claramente identificada em um ambiente de cursos.

8 REFERÊNCIAS

- CASATI, F., CERI, S., PERNICI, B., POZZI, G. *Conceptual Modeling of Workflows*. Proceedings of ER'95, Springer Verlag, Gold Coast, Au., 1995.
- CASATI, F., PERNICI, B. *A Methodology for the Design of WWW Sites and its Application to Distance Education*. Proceedings of SEBD' 96, Pisa, Italy, 1996.
- CASATI, F., CERI, S., PERNICI, B., POZZI, G. *Workflow Evolution*. Proceedings of ER'96, Springer Verlag, Cottbus, Germany, Oct. 1996.
- CASATI, F.; GREFFEN, P.; PERNICI, B.; POZZI, G.; SÁNCHEZ, G. WIDE Workflow model and architecture. Technical Report, ESPRIT Project, 1996.
- CASATI, F.; CERI, S.; PERNICI, B.; POZZI, G. *Workflow Evolution*. Data & Knowledge Engineering 24, 1998, pp. 211-238.
- HAN, Y., SHETH, A., BUSSLER, C. *A Taxonomy of Adaptive Workflow Management*. Proceedings of CSCW'98, Workshop on Adaptive Workflow Systems, Seattle, WA, Nov. 1998.
- HORN, S., JABLONSKI, S. *An Approach to Dynamic Instance Adaption in Workflow Management Applications*. Proceedings of CSCW'98, Workshop on Adaptive Workflow Systems, Seattle, WA, Nov. 1998.
- KUNDE, G. F.; OLIVEIRA, J. P. *Evolução do Esquema Conceitual de um curso a Distância em Workflow*. Workshop Internacional sobre Educação Virtual - WISE'99, Dez 9-11, 1999.
- NICOLAO, M., VICCARI R., EDELWEISS, N., PALAZZO M. DE OLIVEIRA, J.; *Modelagem Conceitual de Workflow para Cursos na Internet*; Anais do IX Simpósio Brasileiro de Informática Educativa, SBC, publicação em CD, SBIE '98, Fortaleza, CE, 17-19 de Novembro 1998.
- PALAZZO M. DE OLIVEIRA, J., NICOLAO M., EDELWEISS, N.; *Conceptual Workflow Modelling for Remote Courses*, Proceedings of IFIP World Computer Congress, Teleteaching '98 Distance Learning, Training and Education, Proceedings of the XV World Computer Congress, IFIP, 31 de agosto a 4 de setembro de 1998, Viena - Áustria e Budapeste - Hungria, p. 789-797.
- REICHERT, M., HENSINGER, P., DADAM, P., *A Framework for Dynamic Changes in Workflow Management Systems*. Proceedings 8th Int. Workshop Database and Expert Systems Applications – DEXA'97, Toulouse, France, Sept. 1997. P. 42-48.
- REICHERT, M., HENSINGER, P., DADAM, P., *ADEPT_{flex} – Supporting Dynamic Changes of Workflows Without Losing Control*. Technical Report No. 97-07, University of Ulm, 1997.
- REICHERT, M., HENSINGER, P., DADAM, P., *Supporting Adaptive Workflows in Advanced Application Environments*. EDBT Workshop on Workflow Management Systems, Valencia, March 1998.
- SIZILIO, G. R. M. A. *Técnicas de Modelagem de Workflow Aplicadas à Autoria e Execução de Cursos de Ensino à Distância*. PPGC da UFRGS- Dissertação de Mestrado, Jan, 2000.
- WORKFLOW MANAGEMENT COALITION. *Terminology & Glossary*. Bruxelas: WfMC, 1996.