

Utilização de Grafo de Alcançabilidade para a verificação de *Design* Instrucional bem formado

Isabel Dillmann Nunes, Ulrich Schiel

Departamento de Sistemas e Computação – Universidade Federal de Campina Grande
Caixa Postal 10.106 – 58.429-900 – Campina Grande – PB – Brasil

inunes@copin.ufcg.edu.br, ulrich@computacao.ufcg.edu.br

Abstract. *Planning and application of sequencing of activities are tasks performed during the Instructional Design, essential in a distance course. Such planning should take into consideration the paths that a student runs in order to achieve the necessary learning. However, some situations can occur which deviates from the conventional course of activities such as activities that do not need to be performed or paths that are not used. The objective of this work is to present the proposal for verification of a well-formed Instructional Design through the use of the Reachability Graph applied to High Level Activities Net (HLAN).*

Resumo. *O planejamento e a aplicação do sequenciamento de atividades são tarefas realizadas durante o Design Instrucional, essenciais em um curso à distância. Tal planejamento deve levar em consideração os caminhos que um aluno percorre com o intuito de alcançar o aprendizado necessário. Porém, algumas situações podem ocorrer que se desviam do percurso normal do aprendizado, como atividades que não precisam ser realizadas ou caminhos que não são utilizados. O objetivo deste trabalho é apresentar a proposta de verificação de um Design Instrucional bem formado através da utilização do Grafo de Alcançabilidade aplicado à notação de Rede de Atividades de Alto Nível (RAAN).*

1. Introdução

Educação a Distância (*E-learning*) é a união entre tecnologia e educação e possui o objetivo de fornecer conhecimento às pessoas através de ferramentas baseadas na *web*. Os Ambientes Virtuais de Aprendizagem – AVA são responsáveis por disponibilizar tal conhecimento às pessoas que estão geograficamente distantes ou com indisponibilidade de horário.

O sucesso do aprendizado disponibilizados nos AVAs depende de alguns fatores [Fetaji and Fetaji, 2010], tais como: planejamento, *design* (questões de personalização, técnicas pedagógicas e instrucionais são especificados), implementação, avaliação e análise.

Nota-se a importância da fase de *design* instrucional, por ser determinante na aplicação do conhecimento e por considerar as abordagens pedagógicas existentes [Filatro, 2010]. Segundo Dallacosta et all. (2010), *Design* Instrucional está relacionado com o planejamento, desenvolvimento, implementação e a avaliação da aprendizagem.

Assim, notações como sistemas de *workflow* aplicados a Educação possibilita tanto o planejamento como o acompanhamento sejam realizados de forma interativa. Porém, como mostra o trabalho de Spoelstra et al. (2008), é necessário criar mecanismos de flexibilidade, como ações atômicas que são pontos de mudanças permitidas como também identificar caminhos mal formulados contendo, por exemplo, atividades que não precisam ser finalizadas ou caminhos que não são finalizados.

As Redes de Petri de Alto Nível podem ser utilizadas para suprir essa necessidade. As redes possuem uma base matemática e formal que permite o desenvolvimento e a simulação de sistemas [Pádua et. al., 2002]. Mais precisamente, as Redes de Petri Coloridas [Jensen, 1997], permitem que atividades sejam modeladas, restrições de caminhos sejam especificadas e que marcas (*tokens*) sejam individualizadas, criando um ambiente propício para o planejamento e acompanhamento de atividades realizadas por alunos em Ambientes Virtuais de Aprendizagem.

A Rede de Atividades de Alto Nível – RAAN é uma notação que traz as características necessárias para atender aos requisitos do planejamento de *Design Instrucional* e seu acompanhamento em tempo de execução, baseada em Redes de Petri Colorida [Nunes e Schiel, 2011].

A partir de um modelo criado em RAAN é necessário que características de bom planejamento sejam verificadas. Este artigo tem o objetivo de propor a utilização do Grafo de Alcançabilidade [Pommereau et all, 2007] para tal finalidade.

O Grafo de Alcançabilidade, conceito oriundo de Redes de Petri, realiza a verificação de um DI bem formado, possibilitando identificar atividades que não precisam ser finalizadas como também situações de *deadlock* (em que um caminho não chega ao fim).

O artigo está organizado da seguinte forma: a seção 2 mostra os conceitos sobre Rede de Atividades de Alto Nível; a seção 3 mostra a construção de DI com RAAN; a verificação de um DI bem formado através do Grafo de Alcançabilidade é discutida e exemplificada na seção 4 e por fim as conclusões e trabalhos futuros são apresentados na seção 5.



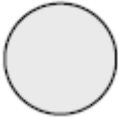



2. Rede de Atividades de Alto Nível (RAAN)

A Rede de Atividades – RA é um modelo gráfico inicialmente usado para planejar e acompanhar o processo de desenvolvimento de um projeto. Ela é uma rede com notação mais expressiva do que Redes de Petri convencionais, pois contém elementos de representação de atividades compostas, eventos e repositório de artefatos, além de cada atividade possuir atributos de custo e tempo [Farias, 2008]. Assim como uma Rede de Petri ela representa um grafo dirigido bipartido e, além disso, esse grafo deve ser uma ordem parcial com um único nó de partida, chamado de *begin* e um nó de finalização, chamado de *end*.

Em analogia a Redes de Petri de Alto Nível, as Redes de Atividades de Alto Nível permitem modelar a execução de uma rede por diversos elementos individualizados sujeitos a restrições de transição. RAs de Alto Nível podem ser aplicadas a qualquer processo dirigido a um objetivo a ser alcançado, ou seja, que visa chegar a um final

específico. Permitem modelar um processo realizado por diversos indivíduos ou grupos simultaneamente. Nesse caso, diversas propriedades que antes eram associados a uma atividade como um todo agora são individualizadas para cada marca que a está executando.

Os elementos gráficos definidos são:

	<p>Atividade básica – uma marca no círculo interno indica que a atividade está sendo realizada; uma marca no círculo externo indica uma atividade encerrada, habilitando transições para outras atividades que dependem dela. Um elemento sem marca indica que a atividade está desabilitada.</p>
	<p>Atividade composta – é uma atividade realizada por uma subrede associada a ela. Uma marca no círculo interno indica que a atividade pode ser iniciada. Uma marca no círculo intermediário indica que a subrede está sendo executada e uma marca no círculo externo indica que a sub-rede encerrou suas atividades.</p>
	<p>Artefato – repositório que armazena os artefatos produzidos pelas atividades e que podem ser utilizados pelas atividades posteriores.</p>
	<p>Transição – é a transição entre as atividades (ou sub-redes) encerradas e as atividades posteriores. Possui uma função cd: $T \rightarrow CG$ (condição de guarda), que associa a cada transição expressões booleanas, a serem avaliadas para habilitação da transição.</p>
	<p>Estabelece ligações entre atividades, eventos, artefatos e transições.</p>
	<p>Evento – permite modelar as condições necessárias para o acionamento de uma transição.</p>
<p>a..z</p>	<p>É uma marca de um indivíduo da rede. Determina o indivíduo que está executando uma atividade e o seu estado nesta atividade (a fazer, em execução, finalizada, executando sub-rede). O estado executando sub-rede destina-se somente para as atividades compostas.</p>
<p>Gi</p>	<p>É uma marca que indica um grupo de indivíduos relacionados entre si e que realizam uma atividade de cooperação. Determina o grupo que está executando uma atividade e o seu estado nesta atividade (a fazer, em execução, finalizada, executando sub-rede). O estado executando sub-rede destina-se somente para as atividades compostas.</p>

As Condições de Guarda devem conter somente referências às propriedades de um indivíduo ou grupo, como por exemplo, nota ou data de finalização de uma atividade. Assim, as Condições de Guarda podem conter os seguintes operadores e funções:

- Operadores lógicos: AND (e), OR (ou) e NOT (negação);
- Operadores relacionais: = (igual), \neq (diferente), $>$ ou \geq (maior ou maior igual), $<$ ou \leq (menor ou menor igual);
- IN(R): neste caso, o predicado da transição está condicionado à inserção (IN) de um documento no repositório por um grupo ou um indivíduo.

As atividades podem ser classificadas em dois tipos:

- Atividade *individual*: identifica o indivíduo que está executando a atividade;
- Atividade *em grupo*: identifica o conjunto de alunos participantes do grupo que executa a atividade. O grupo pode ser formado somente para esta atividade, chamado de *único* e se dissolve no momento em que a atividade é encerrada ou se manter após execução da atividade, chamado de *permanente*.

3. Construção de DI com RAAN

Design Instrucional pode ser visto como uma estrutura gráfica (grafo) composta por uma sequência de atividades com dependências entre si. As relações de dependências podem ser: sequencial, paralelas ou alternativas. Além disso, um DI pode ser considerado como uma sequência que possui um único início e um único fim. Assim, a construção, simulação e acompanhamento de execução de um curso por um *Design Instrucional* (DI) pode ser modelado por uma Rede de Atividades de Alto Nível.

A Figura 1 mostra um exemplo de utilização de predicado que considera os operadores lógicos maior ou igual (\geq) e menor ($<$) como condição para a escolha do caminho que o aluno deve seguir. A atividade “Fazer lista de exercícios” possui como propriedades o identificador de um indivíduo x , nota n e tempo de execução da atividade t . E a atividade “Realizar atividade auxiliar” possui como atributos, além do identificador do indivíduo x , o tempo de execução da atividade t . Os predicados verificam a nota do aluno x para indicar o caminho a ser seguido.

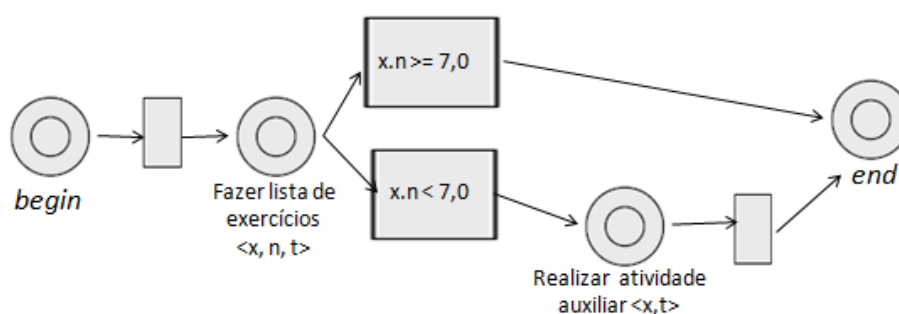


Figura 1. Exemplo de Rede de Atividade com operadores maior ou igual e menor.

O exemplo que verifica o tempo de execução de uma atividade e sua realização pode ser visto na Figura 2. Nesse exemplo, é verificado se o aluno entregou o resumo através da função IN, resultante da atividade “Escrever resumo” e identificada por y e se

essa tarefa foi realizada em um prazo igual ou menor do que 2 dias. A transição que verifica o resultado da atividade “*Escrever resumo*”, mesmo que apresente somente uma restrição em relação a esta atividade somente pode ser habilitada depois de ser finalizada a atividade “*Fazer lista de exercícios*” também. Essa condição deve-se ao conceito de Redes de Petri que define a habilitação de uma transição somente depois do término da realização de todas as atividades anteriores ligadas a ela.

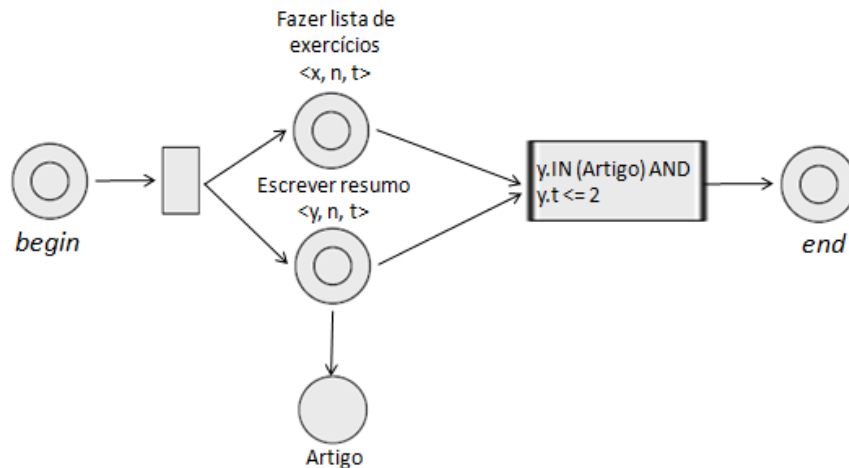


Figura 2. Exemplo de utilização da função IN da verificação de tempo de entrega da atividade.

4. Verificação de um *Design Instrucional* bem formado

Um *Design Instrucional* pode ser considerado bem formado quando a Rede de Atividades de Alto Nível que o representa não contém certas conexões que podem comprometer a sua execução. Formalmente, toda execução da rede com uma marcação em *begin* deve terminar com um única marca em *end*.

As situações em que uma Rede de Atividades pode ser comprometida são: quando não é possível continuar a execução da rede - *deadlock* (Figura 3) e quando uma atividade se torna inútil (Figura 4).

No primeiro exemplo (Figura 3) somente uma das transições (T0 ou T1) é ativada, enquanto que para continuar a execução, T2 espera a execução das atividades P1 e P2. Na Figura 4 as atividades marcadas por T0 habilitam cada um uma transição distinta (T1 e T2). O acionamento de uma delas é suficiente para dar continuidade ao processo sendo que a outra se torna inútil.

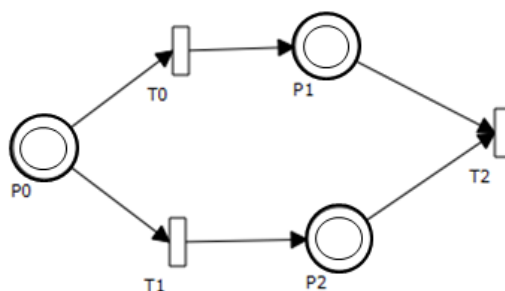


Figura 3. *Deadlock*.

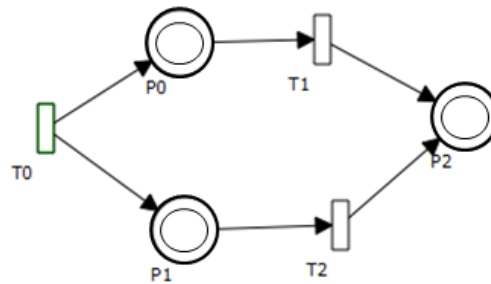


Figura 4. Atividade não finalizada

O Grafo de Alcançabilidade (*Reachability Graph*) permite verificar se uma Rede de Atividades é viva, ou seja, se a rede chega ao fim (não ocorra *deadlock*) ou se não deixa atividades não finalizadas. Um Grafo de Alcançabilidade consiste em um grafo direcionado em que cada nó é uma marcação [Pommereau et al., 2007] e cada aresta representa uma transição. O grafo mostra uma sequência de marcações, em que cada posição de uma marcação corresponde a uma atividade da Rede de Atividades de Alto Nível. Uma marcação representa um momento de execução da rede, onde cada posição identifica uma atividade.

O Algoritmo que gera o Grafo de Alcançabilidade a partir de uma Rede de Atividades de Alto Nível pode ser visto na Figura 5. Considera-se somente uma marca (o que representa somente um aluno no *Design Instrucional*) que percorre a rede e chega ao seu final. Os elementos Repositório e Evento são desconsiderados na construção do Grafo, por não representar uma ação que o aluno irá executar.

1. Rotule a marcação inicial M0 de **raiz** e a sinalize como **nova**;
2. Para cada marcação M **nova**, faça:
 - a. Verifique se a marcação M **nova** já existe no grafo. Se sim, marque M como **velha**;
 - b. Se nenhuma transição estiver habilitada a partir de M, marque M como **fim**;
 - c. Enquanto existir transição t habilitada em M, para cada t, faça:
 - i. Obtenha a marcação M' do disparo de t em M;
 - ii. Introduza M' como nó do grafo, crie o arco com o rótulo t de M para M' e marque M' como **nova**.
 - d. Marque M como uma marcação **visitada**.

Figura 5. Algoritmo para geração do Grafo de Alcançabilidade a partir de uma Rede de Atividades.

Uma Rede de Atividades não será bem formada quando:

- Quando a última marcação do Grafo de Alcançabilidade mostra uma ou mais atividades marcadas, sendo que nenhuma das atividades é a *end*, representando o DEADLOCK (não foi possível chegar ao fim da RA);

- Quando a última marcação do Grafo de Alcançabilidade mostra a atividade *end* marcada, porém com outras atividades marcadas também, representando ATIVIDADES NÃO FINALIZADAS.

4.1 Exemplos de Grafo de Alcançabilidade

A Figura 6 mostra o Grafo de Alcançabilidade do *Design* Instrucional representado pela Rede de Atividades de Alto Nível da Figura 2. Neste caso, tem-se 4 posições representando as 4 atividades da Rede de Atividades da Figura 2, em sequência. Por exemplo, a primeira marcação (1, 0, 0, 0), chamada M0, representa que um aluno está em “*begin*”, enquanto que a segunda marcação (0, 1, 1, 0) representa que o aluno está realizando as duas atividades seguintes de forma paralela “*Fazer lista de exercícios*” e “*Escrever resumo*” (o paralelismo é identificado por mais de uma atividade marcada em uma mesma marcação). Já a última marcação (0, 0, 0, 1), mostra que o aluno chegou ao fim, com somente “*end*” marcado.

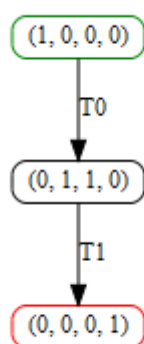


Figura 6. Grafo de Alcançabilidade do DI da Figura 2.

A primeira marcação mostra somente a primeira atividade – *begin* - marcada e a última marcação possui somente a última posição – *end* – marcada, mostrando que DI possui uma execução que permite o aluno finalizar o curso. Assim, o Grafo de Alcançabilidade deste exemplo, mostra que não houve *deadlock*, pois a execução chegou ao final e também que não houve atividades não finalizadas, pois na última marcação somente a última posição está marcada (última marcação com somente a atividade *end* marcada).

Já o exemplo da Figura 7, mostra um *Design* Instrucional mais elaborado que não é bem-formada. Após a atividade “Ler artigo de apoio” o aluno pode escolher entre “Interagir Chat” para tirar suas dúvidas ou “Fazer exercícios”. Porém, a transição T6 espera que as duas atividades anteriores sejam realizadas. Como essa situação nunca irá ocorrer (“Interação chat” e “Fazer exercícios” são atividades alternativas), caso o aluno realize esse caminho, ele não conseguirá atingir o final do curso.

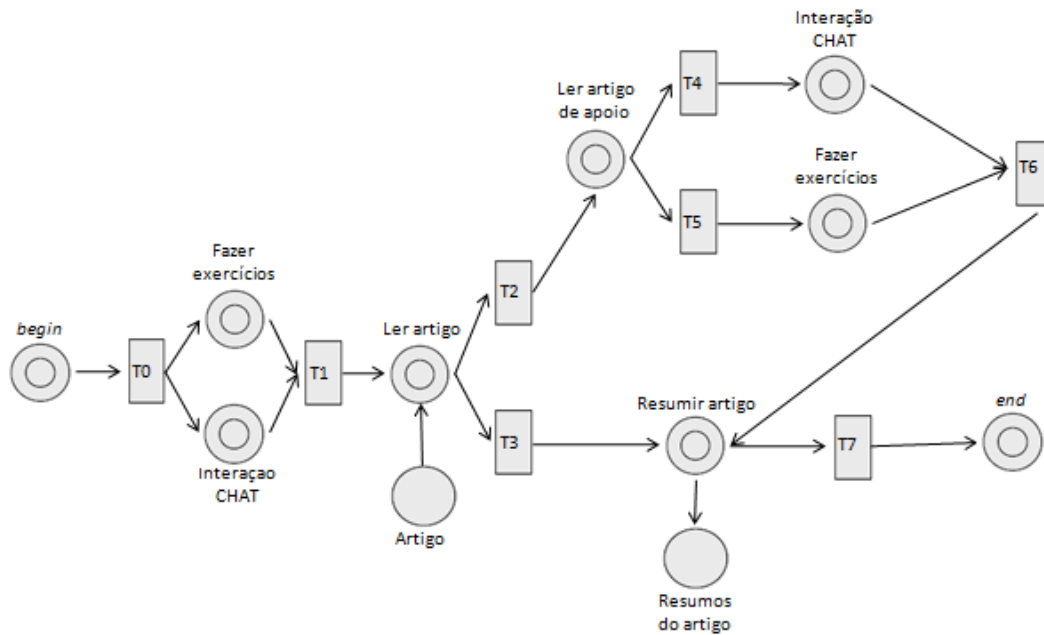


Figura 7. *Design Instrucional* representado por uma Rede de Atividades de Alto Nível com *deadlock*.

A Figura 8 mostra o Grafo de Alcançabilidade do exemplo da Figura 7. O Grafo mostra 3 finais. O primeiro, caso o aluno não necessite “Ler o artigo de apoio”, termina corretamente (com apenas a última atividade *end* marcada) como mostra a marcação (0, 0, 0, 0, 0, 0, 0, 0, 1).

Já as duas outras situações de finalização do Grafo de Alcançabilidade mostram que caso o aluno realize a atividade “Ler o artigo de apoio” acontece o encerramento do curso na atividade “Interação chat” (0, 0, 0, 0, 0, 1, 0, 0, 0) ou “Fazer exercícios” (0, 0, 0, 0, 0, 1, 0, 0). Assim, é possível verificar que o *Design Instrucional* foi mal planejado e é preciso reformulá-lo para que o curso, em todas as suas possibilidades, possa ser finalizado.

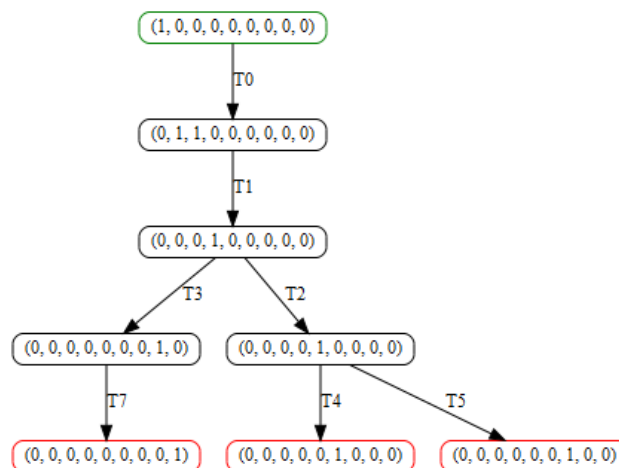


Figura 7. Grafo de Alcançabilidade com *deadlock* do DI da figura 4.

5. Conclusão

O planejamento de um curso a distância depende de vários fatores, entre eles o seu *Design* Instrucional. O DI deve ser modelado de forma que os perfis dos alunos sejam levados em consideração, a metodologia de educação utilizada seja respeitada e a sequência de atividades esteja em conformidade com tais definições.

A utilização de Rede de Atividades de Alto Nível para a construção do DI permite a utilização de uma notação gráfica e de fácil assimilação, prevendo atividades compostas que devem ser divididas em uma sub-rede, criação de artefatos e sua utilização em outras atividades e controle de atividades por eventos que são modelados.

Além disso, a RAAN identifica caminhos do *Design* Instrucional que não foram bem planejados através da geração do Grafo de Alcançabilidade correspondente. Permitindo assim reconhecer atividades que não precisam ser finalizadas como também caminhos que não chegam ao fim. A partir da identificação de tais problemas é necessário aplicar o modelo a cursos reais de Educação a Distância com o intuito de reconhecer as reais contribuições do trabalho.

Para que o modelo seja aplicado é necessário a implementação da ferramenta ATID (*Authoring Tool for Instructional Design*) que realiza tanto a construção e acompanhamento do *Design* Instrucional como também a sua verificação. A ATID está sendo desenvolvida por um grupo de alunos através de seus trabalhos de conclusão de curso e iniciação científica [Paraíba et al, 2012].

A ATID permitirá sua integração com qualquer Ambiente Virtual de Aprendizagem. Primeiramente está sendo desenvolvida sua integração com o ambiente Moodle através de Agentes Inteligentes. Tal integração permitirá que o modelo construído na notação de Rede de Atividades de Alto Nível seja exportado para o Moodle e qualquer modificação durante a execução do curso (tanto no próprio Moodle quanto na ATID), será automaticamente reportada para o outro ambiente.

Referências Bibliográficas

- Dallacosta, A.; Cazetta, G.; Souza, S. G de. (2010). “Novas tecnologias aplicadas na elaboração de material instrucional online”. 3º Simpósio Hipertexto e Tecnologias na Educação (redes sociais e aprendizagem). Recife - PE.
- Farias, C. B. A. (2008). “Uma extensão de Rede de Petri para Modelagem de Processos e Controle de Projetos”. Tese de Doutorado. Orientador: Ulrich Schiel. Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande. 281 páginas.
- Fetaji, B; Fetaji, M. (2010). “E-Learning Indicators: A Multidimensional Model for Planning Developing and Evaluating E-Learning Software Solutions”. In: E-Learning, experiences and future. Edited by Safeullah Soomro and published by In-Teh. Pag 1-34.
- Filatro, A. (2010). “Design Instrucional Contextualizado – educação e tecnologia”. 3ª edição. Editora Senac – São Paulo. 215 páginas.

- Jensen, K. (1997). "Coloured Petri Nets – Basic Concepts, Analysis Methods and Practical Use". Volume 1. Second Edition. Springer-Verlag Berlin Heidelberg New York.
- Nunes, I. D.; Schiel, U. (2011). "Design Instrucional e seu acompanhamento em tempo de execução utilizando Rede de Atividades". 22º Simpósio Brasileiro de Informática na Educação - SBIE. Aracajú – SE.
- Pádua, S. I. D. de; Silva, A. R. Y. da; Inamasu, R. Y.; Porto, A. J. V. (2002). "Redes de Petri aplicadas aos Sistemas de Gerenciamento de Workflow". XXII Encontro Nacional de Engenharia de Produção. Curitiba – PR. 2002.
- Paraíba, H. L. P.; Escorel, J. B.; Vieira, P. A. P.; Britto, R.; Soares, W. L. F.; Nunes, I. D.; Schiel, U. (2012). "ATID – Authoring Tool for Instructional Design". International Free Software Workshop. In: 13º FISL – Fórum Internacional de Software Livre. Porto Alegre – RS. 2012.
- Pommereau, F.; Devillers, R.; Klaudel, H. (2007). "Efficient reachability graph representation of Petri nets with unbounded counters". Published by Elsevier Science B. V.
- Spoelstra, H.; Matera, M.; Rusman, E.; Bruggen, J. van.; Koper, R. (2008). "Bridging the gap between instructional design and double loop learning". International Journal of Web-Based Learning and Teaching Technologies (IJWLTT). Volume 3, Issue 1. 12 pp.