

Uma Extensão de Rede de Petri para Modelagem de Processos e Controle de Projetos

Cecir Barbosa de Almeida Farias¹, Ulrich Schiel²

¹Departamento de Engenharia Elétrica

²Departamento de Sistemas e Computação

Universidade Federal de Campina Grande (UFCG) – Campina Grande, PB – Brasil

{cecir@dee.ufcg.edu.br, ulrich@dsc.ufcg.edu.br}

***Abstract.** Complex projects are developed with difficulty, although there exists techniques, and tools for project management. Deadlines always are extended and the cost of software being developed must be incremented. The solution proposed in this article is based in the use of an extension of Petri Nets for project management called Activities Net, and designed to track detailed information about the development of processes and projects. This net is the main layer of the Integrated System of Project Development Management. In order to determine the structural quality of an Activities Net and allow simulations and correct cost a time calculations, a special class of Petri Nets has been developed, called Project Net.*

***Resumo.** Projetos complexos são desenvolvidos com dificuldade, apesar de existirem técnica e ferramentas de gestão de projetos. Prazos são estendidos e o custo do software acaba sendo incrementado. A solução proposta neste artigo tem como base o uso de uma extensão de Redes de Petri para gerenciamento de projetos chamada Rede de Atividades, utilizada para detalhar informações sobre o desenvolvimento de processos e projetos. Esta rede é o componente principal de um sistema integrado de gerência de desenvolvimento projetos. Para determinar a qualidade estrutural de uma Rede de Atividades permitir simulações e cálculos dos custos e prazos, foi desenvolvido um tipo especial de rede de Petri, denominado Rede de Projeto.*

1. Introdução

Um projeto consiste em uma proposição de realização de certo objetivo pré-definido. A realização dele consiste na execução de uma série de atividades com a finalidade de se conseguir este objetivo, que pode ser chamada de produto do projeto. Portanto, esta execução tem um início e um fim, tem um orçamento a ser executado, uma equipe formada por um coordenador e desenvolvedores e possui recursos necessários para construir o produto. A forma de executar o projeto segue um processo bem definido, chamado de metodologia de desenvolvimento. Dependendo do tipo de projeto, deverá ser adotada a metodologia adequada.

Os modelos tradicionais de gestão de projetos, tais como o método do caminho crítico, o método de Gantt, e os métodos de avaliação gráfica e revisão técnica (PERT/CPM) (Cukierman, 2001) são utilizados para gerenciar projetos, mas não consideram se os recursos estão disponíveis e não permitem modelar situações dinâmicas (Meredith e

Mantel, 2000).

Redes de Petri (Murata, 1989), por outro lado, são conhecidas pela sua capacidade em modelar atividades concorrentes e simular a evolução dos processos. A gestão de projetos tem sido identificada por alguns pesquisadores como uma área próspera onde Redes de Petri podem ser usadas.

O trabalho desenvolvido, e apresentado neste artigo, tem como contexto o acompanhamento do desenvolvimento de projetos, em particular, projetos de software. Pretende-se contribuir para o gerenciamento de processos e de projetos, em particular, no controle de custos e prazos. É utilizada uma estrutura baseada em Redes de Petri denominada Rede de Atividades - RA (Torres, 1996) para modelar e acompanhar as atividades de um processo em desenvolvimento e suas interdependências funcionais.

Com a Rede de Atividades é possível realizar a modelagem das etapas de metodologias de desenvolvimento de software e definir atividades. Ela foi estendida neste trabalho, para permitir modelar, além do acompanhamento detalhado do processo de desenvolvimento e interdependência de todas as atividades, os respectivos custos, prazos e recursos alocados, calcular tempos globais e confrontar continuamente os dados reais com os dados previstos.

Para garantir a composição precisa das atividades para calcular custos e prazos globais, foi criado um tipo especial de rede de Petri, denominada Rede de Projetos, que permite analisar as características de uma Rede de Atividades, modelando-a da maneira mais adequada para cálculos de custos e prazos.

Além disso, foi proposto um ambiente de desenvolvimento de projetos denominado GDP - Gerência do Desenvolvimento de Projetos.

1.1. Trabalhos Relacionados

Existem alguns esforços para uso de redes de Petri para planejamento e alocação de recursos no gerenciamento de projeto e modelagem de sistemas complexos.

Os trabalhos de (Chen, Hsu e Chang, 2008) e (Meredith e Mantel, 2000) apresentam soluções para modelar o controle de recursos e a divisão de tarefas. O trabalho de Wang, Xuw e Yang (2007) apresenta um modelo de rede de Petri como modelo de *workflow* de gerenciamento direcionado apenas à pesquisa científica.

Observa-se que, estas extensões de redes de Petri tratam de características específicas do gerenciamento de projeto (tempo e recurso, ou só tempo) atuando apenas com um foco isolado e não foram desenvolvidas para gerenciamento de processos de desenvolvimento de software (XP, RUP, etc.).

Existem ferramentas que auxiliam o processo de desenvolvimento de software tais como: Rational Unified Process Builder (2004), XPlanner (2005) e Microsoft Visual Studio Team System (VSTS, 2008). Elas possuem funcionalidades para acompanhamento dos processos RUP (Jacobson, I.; Booch, G.; Rumbaugh, J., 1999), XP (2003) e MSF (Vianna, 2004) respectivamente. Mas não permitem o uso para vários tipos diferentes de processos e não controlam todas as variáveis de projeto (custo, tempo e recursos) (Farias, 2007).

Existem outras ferramentas, por exemplo, GENESIS (Aversano, 2004), ODE (2005) e APSEE (2004). GENESIS é utilizada para a modelagem de processo, cabendo

ao gerente mapeá-lo usando uma estrutura de formulários e não um diagrama visual. As redes que são usadas nestas ferramentas não seguem um modelo formal de Redes de Petri, elas foram criadas especificamente para as ferramentas e não provêem recursos que permitam o acompanhamento do processo, confrontando o planejado com o realizado, necessário para a criação de uma ferramenta real de controle de projetos.

Dentre as ferramentas comerciais, por exemplo: Microsoft Project (2005) e MinuteMan (2003) permitem descrever tarefas, custos, geram relatórios de cronogramas, gráficos de barras, e permitem o compartilhamento do trabalho entre gerente e equipes. Elas coletam, agrupam e organizam dados do projeto, mas não permitem modelagem e acompanhamento de metodologias de desenvolvimento de software.

O que se observa é que muitas vezes são necessárias duas ou três ferramentas dentre as citadas, em um mesmo projeto. Em TABA (2004) e em dotProject (2007), por exemplo, não há nenhum recurso visual todas as etapas do **processo** de desenvolvimento e todos os dados do **projeto** (atividades, tempos, custos e recursos) realizando totalizações e simulações, simultaneamente.

Na seção 2 são apresentadas as Redes de Atividades e Rede de Projetos. Na seção 3, é apresentado o ambiente GDP - Gerência de Desenvolvimento de Projetos, arquitetura e módulos. Na seção 4 é apresentado um exemplo de uso das Redes de Atividades e de Projetos no GDP e definição de atividades, cálculos de custos e tempos. Na seção 5 é apresentada a forma de validação. Por fim, são apresentadas as conclusões.

2. Rede de Atividades (RA) e Rede de Projeto (RP)

Nesta seção, são apresentadas as Redes de Projetos e Redes de Atividades. A Rede de Atividades (Torres, 1996) é uma estrutura que apóia a gerência de projeto de software e é usada para acompanhar o processo de desenvolvimento de um projeto. Ela é uma rede com uma notação gráfica mais expressiva do que as Redes de Petri (Murata, 1989) convencionais, pois possui elementos de representação de atividades compostas, eventos e repositórios de artefatos.

2.1. Rede de Atividades – Definição Formal

Uma Rede de Atividades Básica é uma estrutura $ra = \langle A, T, F, E, \alpha, \beta, tp, tr, cp, cr \rangle$:

$A = AT \cup AF$ são atividades (AT) ou artefatos (AF);

$AT = AB \cup AC$ são atividades básicas (AB) ou compostas (AC);

T é um conjunto de transições;

E é um conjunto de eventos;

$F \subseteq AT \times T \cup T \times AT \cup AT \times AF \cup AF \times AT \cup E \times T \cup T \times E$ é um conjunto de arestas; $\alpha: (AT \cup E) \rightarrow \{d, e, f, s\}$ é a função que determina os estados possíveis das atividades, podendo estar *desativada*, *em execução*, *finalizada* ou *executando sub-rede*, respectivamente. O estado *executando sub-rede* só se aplica a elementos de AC (atividades compostas).

$\beta: E \rightarrow \{0, \dots, n\}$ é a função de marcação de eventos

$tp: AT \rightarrow \text{Tempo}$, que associa a cada atividade o tempo previsto

$tr: AT \rightarrow \text{Tempo}$, que associa a cada atividade o tempo real

$cp: AT \rightarrow \text{Custo}$, que associa a cada atividade o custo previsto

cr: $AT \rightarrow$ Custo, que associa a cada atividade o custo real

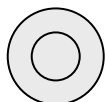
Uma transição está habilitada quando as atividades de entrada estão *finalizadas* ($\alpha(a_i)=f$), os eventos de entrada estão marcados e todas as atividades de saída estão *desativadas* ($\alpha(a_i)=d$).

O *acionamento* de uma transição habilitada consiste em:

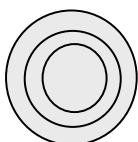
- Alterar os estados das atividades de entrada de *finalizada* para *desativada*;
- Desmarcar os eventos de entrada;
- Marcar as atividades de saída como *em execução*; marcar os eventos de saída.

2.1.1. Visualização gráfica da Rede de Atividades

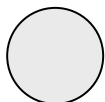
Uma Rede de Atividades pode ser visualizada com os seguintes elementos gráficos:



Atividade básica - com uma marcação no círculo interno, a atividade está *sendo executada*; marcação no anel externo e sem marcação no anel interno significa atividade *encerrada*, habilitando a transição para outras atividades que dependem desta; sem marca, significa que a atividade está desativada.



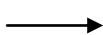
Atividade composta - atividade realizada por uma sub-rede associada a ela. Marcação no anel interno significa que a sub-rede pode ser iniciada; marca no círculo intermediário significa que a sub-rede esta sendo executada; marcação no anel externo significa que a sub-rede encerrou suas atividades, habilitando a transição para outras atividades.



Artefato - repositório que armazena artefatos produzidos pelas atividades e que poderão ser utilizados pelas atividades posteriores.



Transição - representa a transição entre atividades encerradas e novas atividades, dependendo das anteriores. Se acionadas, as atividades de saída se tornam ativas.



Uma aresta dirigida permite estabelecer ligações entre atividades, eventos, artefatos e transições.



Uma marca preta representa uma marcação na rede. Ela determina o estado da atividade (em execução, desativada ou encerrada)



Evento - permite modelar condições ou circunstâncias necessárias para o acionamento de uma transição.

Versões estendidas de Redes de Atividades consideram diversas marcas em uma mesma atividade, significando execuções repetidas de uma mesma atividade. Se uma mesma atividade está sendo executada por diversos indivíduos, eles podem ser identificados por variáveis, tais como 'x', 'y'.

Uma Rede de Atividades permite modelar as principais interdependências que podem existir entre o desenvolvimento de atividades de um projeto. São elas:

- *Execução seqüencial*. Duas atividades a_1 e a_2 têm uma execução seqüencial se existe uma transição t , tal que $\langle a_1, t \rangle$ e $\langle t, a_2 \rangle$ são arestas em F ;
- *Execução paralela*. Duas atividades a_1 e a_2 são independentes se existe uma transição t tal que, $\langle t, a_1 \rangle$ e $\langle t, a_2 \rangle$ estão em F ;

- *Execução alternativa*. Duas atividades a_1 e a_2 são mutuamente exclusivas se existe uma atividade a_0 e transições t_1 e t_2 , tal que $\langle a_0, t_1 \rangle$, $\langle a_0, t_2 \rangle$, $\langle t_1, a_1 \rangle$ e $\langle t_2, a_2 \rangle$ estão em F .
- *Loop*. Uma Rede de Atividades RA tem um loop em uma atividade 'a' se existe um caminho cíclico de 'a' e retorno para 'a'. Neste caso, o grafo RA é cíclico.

2.2. Redes de Projeto

A estrutura de desenvolvimento de um projeto pode ser modelada por uma Rede de Petri. Definimos, neste item, uma classe de Rede de Petri que atende exatamente aos requisitos estruturais de um ambiente de desenvolvimento de projetos. As características específicas desta rede servirão de base para a introdução da Rede de Atividades que permite modelar os processos de desenvolvimento de um produto, com suas interdependências funcionais, para especificação de dados de tempo, custo, recursos humanos e materiais, e artefatos produzidos.

2.2.1. Definição da Rede de Projeto

Em uma Rede de Projeto, os lugares P da rede são chamados de atividades. Por ser dupla livre-escolha, evitam-se referências cruzadas indesejáveis na rede. Os limites do reticulado determinam o começo e o fim do projeto. A última propriedade garante que todo projeto chega ao fim.

Uma Rede de Projeto é uma rede de Petri $N = \langle P, T, F \rangle$, em que:

P é um conjunto de lugares, contendo um lugar *fim* $\in P$

T é um conjunto de transições, contendo uma transição *inicio* $\in T$;

F é um conjunto de arestas com as seguintes propriedades:

- Ela é dupla livre-escolha¹;
- O grafo associado a N é um reticulado² não-vazio com $\text{inf} = \text{inicio}$ e $\text{sup} = \text{fim}$;
- O sistema $\langle N, \emptyset \rangle$ é vivo³;
- $\text{inicio}^* = \emptyset$ e $\text{fim}^* = \emptyset$.

Os limites inferiores e superiores são denominados *inicio*(*inf*) e *fim*(*sup*) do projeto. Deve-se observar que a definição de Rede de Projeto não permite ciclos. Se um projeto real contém execuções repetidas das mesmas, consideramos, para as definições formais, que o número de repetições é fixo e que, portanto, o ciclo pode ser linearizado.

¹ Uma rede $N = \langle P, T, F \rangle$ é chamada **livre-escolha** se para cada arco de P para T $(p, t) \in F$, implica que $|p^*| = 1$ ou $|t^*| = 1$. Chama-se a rede de **dupla livre-escolha** se a mesma propriedade se aplica para todos os arcos (t, p) de T para P , isto é, $|t^*| = 1$ ou $|p^*| = 1$.

² Um **reticulado** é um conjunto finito L parcialmente ordenado com a propriedade de que para todo $a, b \in L$ existem dois elementos de L , chamados $\text{inf}(a, b)$ e $\text{sup}(a, b)$, tal que $\text{inf}(a, b) \leq a \leq \text{sup}(a, b)$, $\text{inf}(a, b) \leq b \leq \text{sup}(a, b)$. Desde que é finito, ele contém um menor elemento, chamado **inf**, e um maior elemento, chamado **sup**.

³ Um sistema $\langle N, M_0 \rangle$ é chamado de **vivo** se para todo lugar $p \in P$ existe um $M \in /M_0$ que permite que seja marcado a cada disparo.

Dado um Sistema $\langle R, M_0 \rangle$, uma seqüência de marcações M_0, \dots, M_k do fecho de M_0 é dita um **caminho** se existem t_1, \dots, t_k tal que $t_i(M_{i-1}) = M_i$ para $i=1, \dots, k$. Esta seqüência $[t_1, \dots, t_k]$ é chamada de um **caminho de transição** de t_1 para t_k .

Deve ser garantido pela rede que todo projeto, em qualquer estado intermediário, chega ao fim. Isto é dado pela seguinte proposição:

Proposição 1: *em uma Rede de Projeto*

- *Todos caminhos maximais são da forma $[\emptyset, \dots, fim]$, e*
- *Todo caminho da rede contido no fecho de \emptyset é um sub-caminho de um caminho maximal.*

Proposição 2: *O grafo do fecho de \emptyset em uma rede de projeto é um reticulado com $inf = \emptyset$ e $sup = fim$.*

Dado um elemento $x \in P \cup T$, distinguimos três casos de evolução do projeto, dependendo da cardinalidade $|x^*|$:

- se, dado um arco (x,y) , temos $|x^*| = |y^*| = 1$, nós chamamos o arco (x,y) de **linear**;
- se $x \in T$ e $|x^*| > 1$, temos uma **bifurcação concorrente** (*concurrent forking*) de grau $|x|$;
- se $x \in P$ e $|x^*| > 1$, temos uma **bifurcação alternativa** (*alternative forking*) de grau $|x|$.

Isto significa que múltiplos arcos, saindo da transição, iniciam uma execução concorrente de diversas atividades, enquanto múltiplos arcos saindo de um lugar (uma atividade) permitem a execução de atividades alternativas mutuamente exclusivas para atingir um mesmo objetivo.

A cada bifurcação em uma rede de projeto deve corresponder, depois na rede, uma **junção** correspondente a fim de fechar o caminho múltiplo. Neste sentido definimos uma **junção de concorrência** como uma transição $t \in T$ tal que $|t^*| > 1$. Analogamente uma **junção de alternativas** é uma atividade $a \in P$ tal que $|a^*| > 1$.

Para garantir uma correlação adequada entre bifurcações e junções deve-se evitar dois casos na rede. Primeiro, devemos evitar blocos mistos, isto é, uma bifurcação concorrente se fechar por uma junção alternativa, ou uma bifurcação alternativa se fechar por uma junção concorrente.

Portanto, definimos uma **Rede de Projeto Bem Estruturada** como uma rede de projeto cujo grafo é bem estruturado e sem bifurcações.

A análise de uma Rede de Projeto bem estruturado pode ser reduzida a analisar cada bloco componente da rede de uma forma *bottom-up*. Assim o cálculo de custos e tempos de um projeto complexo será obtido pela totalização adequada de cálculos em blocos elementares. Um bloco é **elementar** se não houver nenhum bloco dentro dele. Isto significa que um bloco $[b_1, b_n]$ é elementar, se para todo bloco $[c_1, c_n]$ com $b_1 \leq c_1$, e $c_n \leq b_n$ temos $b_1 = c_1$ e $c_n = b_n$.

3. GDP – Gerência de Desenvolvimento de Projetos

O GDP é fornece uma maneira fácil e eficiente para determinar o tipo de produto a ser desenvolvido a fim de escolher a metodologia de desenvolvimento adequada, modelar metodologias de desenvolvimento de software e acompanhar projetos. Apesar de o ambiente ser de propósito geral, é dado um enfoque maior no desenvolvimento de software.

Para realizar seus propósitos o GDP utiliza duas ferramentas ou módulos: o módulo ERA - Editor de Redes de Atividades e o SiSeM - Sistema de Seleção de Metodologias de Desenvolvimento de Software, descritos a seguir.

Toda atividade possui uma descrição, os artefatos gerados ou necessários, as relações de dependência com outras atividades, as datas de início e fim planejadas, os recursos a serem alocados e os agentes responsáveis pela mesma.

Os principais usuários do GDP são o gerente de processos, que irá definir qual o melhor processo a usar para um projeto em particular e modelar a rede do processo; o gerente de projeto, que cria a RA do projeto a ser criado e acompanha seu desenvolvimento e totalização; e a equipe de desenvolvedores, que faz a atualização de tempos na RA do projeto em desenvolvimento.

3.1. Arquitetura do GDP

O GDP dá suporte à gerência do desenvolvimento de projetos em quatro níveis:

Descrevemos, brevemente, cada um destes níveis da arquitetura da Figura 1.

Nível 1 - Escolher metodologia: por meio de um sistema de apoio à decisão, que utiliza uma árvore de decisão, e realiza em um diálogo com o usuário, são definidas as principais características do projeto a ser realizado a fim de determinar a metodologia mais adequada (uso do módulo SiSeM).

Nível 2 - Criar a RA de uma metodologia: o gerente de processo cria uma Rede de Atividades desta sistemática, determinando a interdependência funcional entre as diversas etapas propostas pela metodologia. A RA criada será armazenada na base de metodologias e servir como um modelo (*template*) que pode ser adaptado (customizado) para um projeto concreto a ser desenvolvido segundo a metodologia.

Nível 3 - Criar RA do projeto: O gerente de projetos cria a rede da aplicação a ser desenvolvida (modela os passos) se ela já não estiver armazenada no banco de dados. Se o processo utilizar uma metodologia conhecida, armazenada na base de metodologias, ele utilizará a RA correspondente que será instanciada para o projeto em questão. Serão determinadas as atividades do projeto, e os tempos, recursos e custos estimados.

Nível 4 - Executar projeto: Durante a execução do projeto os desenvolvedores inserem datas reais e custos de conclusão das atividades e o gerente do projeto pode fazer estimativas, acompanhar os tempos e custos reais de um projeto e executar correções no projeto.

3.2. Módulos do GDP

Os dois principais módulos da camada lógica do GDP são: SiSeM e ERA.

a. Módulo SiSeM – o Sistema de Suporte à Escolha da Metodologia é um Sistema de Suporte a Decisão (SSD) interativo que auxilia o gerente de processo para, antes do início do projeto, escolher a metodologia mais adequada ao tipo de sistema a ser desenvolvido. Baseado em uma árvore de decisão o SISEM obtém, por meio de um diálogo, as principais características do sistema a ser desenvolvido. Para cada resposta, cada metodologia recebe uma pontuação (variando entre 0 e dez), determinada previamente de acordo com os estudos realizados e entrevistas realizadas com equipes de desenvolvimento de software.

b. Módulo Editor de Rede de Atividades – ERA - neste módulo a Rede de Atividades pode ser criada, atualizada e executada. O ERA é composto dos seguintes sub-módulos:

- Interface – com este módulo é possível criar uma RA. Ele é um **editor gráfico** e para cada atividade contém: nome da atividade, pré e pós condições de atividades, eventos, artefatos, tempos e custos estimados e desenvolvedores alocados.
- Gerenciamento de recursos - gerencia a alocação de pessoal e equipamentos ao projeto.
- Armazenamento e recuperação da rede - armazena e recupera uma RA da Base de Projeto.
- Simulador - em cada ponto do projeto são calculados o custo global estimado, custo real, custo restante, tempo global estimado, tempo real e tempo restante.

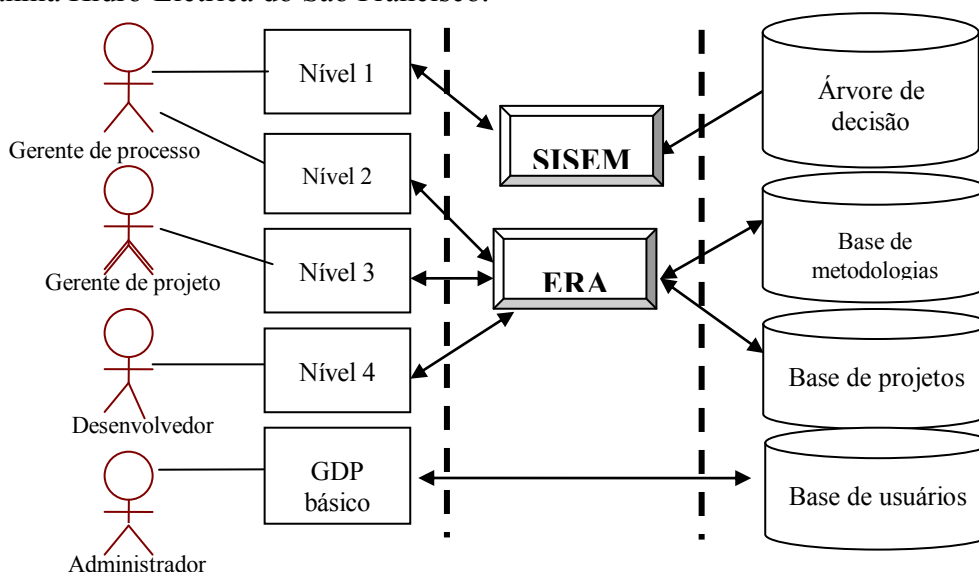
c. Base de projetos – para cada projeto armazena a RA e contém dados de recursos humanos, recursos operacionais e artefatos desenvolvidos.

d. Base de metodologias – contém as Redes de Atividades criadas pelo gerente de processos a ser usada pelo gerente de projetos.

e. Base de usuários – é mantida pelo administrador e contém dados e níveis de autorização dos diversos usuários do sistema.

4. Exemplo de Cálculo de Custos e Tempos do Projeto com o GDP e as redes

A aplicação prática da Rede de Atividades foi realizada com a metodologia XP e com gerente e equipe do projeto SAD - Sistema de Apoio à Decisão (2007) da CHESF - Companhia Hidro-Elétrica do São Francisco.



Camada de apresentação Camada lógica Camada de dados

Figura 1. Arquitetura do ambiente GDP.

Inicialmente, o gerente do projeto utilizou o SiSeM para descobrir a metodologia mais adequada ao seu projeto – Metodologia XP. Em seguida, realizou a criação da rede de Atividades da metodologia XP1 (XP1, 2007) - uma extensão da metodologia XP, com o ambiente GDP, módulo gráfico. O próximo passo foi adaptar a rede criada de acordo com as etapas do planejamento do projeto SAD (Figura 2). As etapas foram definidas pelos “tutores” do projeto em conjunto com gerente do projeto para os dois anos de duração. O gerente de projeto abriu a rede de XP1 e realizou a adaptação da rede de acordo com a documentação do projeto.

Foi realizada também a instanciação da rede incluindo os valores de tempo (Figura 2), custos e definição dos recursos humanos para as diversas atividades do projeto, através de tela de formulário. Após a instanciação dos itens do projeto, os desenvolvedores foram convidados a participar do estudo de caso, em seções individuais.

Cada desenvolvedor observou a Rede de Atividades criada para o projeto e verificou quais as suas funções a realizar. Após a conclusão da atividade eles inseriam os tempos realizados (data e hora) com as atividades Enquanto isso o gerente podia acompanhar a realização das atividades do projeto através do módulo gráfico do editor ERA.

Após a conclusão das atividades, o gerente e sua equipe observaram os totais realizados para a primeira etapa do projeto (aquisição), através do módulo de simulação do ERA (Figura 2).

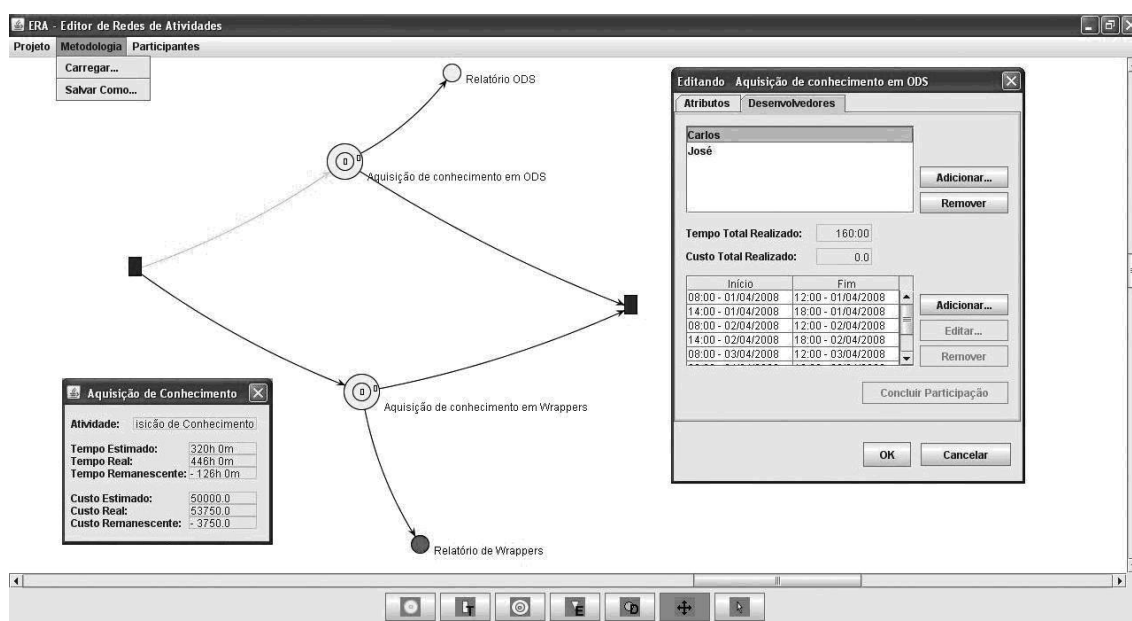


Figura 2. Modelagem da rede do projeto SAD com instanciação de valores e totais de tempo e custo para atividade “Aquisição de conhecimento em ODS”.

5. Validação dos Resultados

Além do uso prático do sistema junto ao projeto SAD, foram realizados experimentos com 30 usuários com perfil semelhante aos usuários do SAD. Foram recrutados gerentes de projetos e alunos no Departamento de Sistemas e Computação da Universidade Federal de Campina Grande nos laboratórios de projetos. Os participantes foram divididos em dois grupos: experientes e inexperientes de acordo com sua experiência prévia, conhecida através dos questionários pré-teste.

O experimento consistiu em analisar a realização de tarefas no sistema GDP pelos usuários, a partir da metodologia de Queiroz (2001). Foram utilizados instrumentos para sondagem da satisfação subjetiva: questionários pré e pós-teste para delineamento do perfil deles enquanto utilizavam o sistema, e adotados os instrumentos para mensuração do desempenho: roteiro de tarefas e fichas de registros de eventos realizados.

Após isto, foram realizadas a tabulação e análise dos dados obtidos a partir dos dois enfoques, assim como o procedimento estatístico destes dados, uma discussão dos problemas registrados e descritas recomendações. O procedimento estatístico mais adequado ao nosso propósito foi o *Teste t* (Barbetta, 2008), que verifica diferenças entre duas categorias, possibilitando comparações entre médias de dois grupos e verificar se há diferenças significativas.

A mensuração do desempenho foi realizada a partir de dados sobre três aspectos do procedimento de teste: tempo de execução das tarefas, número de ações e de escolhas incorretas. Com a visualização da RA, 20% dos usuários verbalizaram que gostariam de visualizar a hierarquia total de níveis e sub-níveis em uma única tela para ver o processo geral. Para isto, poderia ser incluída uma sobreposição de nível na tela, de acordo com a seleção do usuário, ao clicar na atividade desejada.

A partir das respostas dos questionários pós-teste, a Tabela 1 apresenta uma lista de alguns aspectos do processo interativo usuário-produto que apresentaram problemas.

Tabela 1. Sumário de problemas e comentários sobre o ambiente a partir da sondagem da satisfação (questionário pós-teste).

Problema	Item do questionário	Solução para problema encontrado
1	Simulação de totais do projeto.	Revisar o mecanismo de acesso ao início da funcionalidade de <i>simulação</i> .
2	Entendimento e compreensão das instruções <i>online</i> .	Inserir um tipo de ajuda <i>on-line</i> e <i>off-line</i> a fim de tornar mais fácil a superação de dúvidas e problemas encontrados pelo usuário durante o uso do produto.
3	Processo de entrada e saída de dados durante o uso do software.	Revisar a estruturação a forma de apresentação adotada na entrada e saída do software, a fim de tornar mais fácil sua compreensão pelo usuário.

Por fim, houve a confrontação dos resultados obtidos a partir dos dois enfoques adotados, discutindo as similaridades e diferenças resultantes do experimento. Foi

mostrado que os usuários principiantes conseguem executar as mesmas tarefas dos usuários experientes, com baixa diferença de médias aritméticas, com a Rede de Atividades no software editor de RAs, ainda que às vezes em período maior de duração, e números de escolhas incorretas, em especial quando as tarefas exigem experiência e conhecimento prévio.

6. Conclusão

Com as Redes de Atividades e Rede de Projeto, e sistema GDP - Gerência de Desenvolvimento de Projetos é possível cobrir o ciclo de gestão de projetos, desde a sua concepção inicial até sua implantação completa.

O GDP permite o planejamento e acompanhamento de projetos através dos módulos: **SISEM** – Sistema para Seleção de Metodologias de Desenvolvimento de Software e **ERA** – Editor de Rede de Atividades.

O acompanhamento de determinado projeto é apresentado na forma da Rede de Atividades que exibe no módulo ERA, através de seus componentes, o fluxo das atividades, e para cada atividade são mostradas informações tais como: tempos e custos estimados, remanescentes, reais e recursos alocados. Dessa forma, gerente de projeto e desenvolvedores podem acompanhar o andamento dos projetos.

Atualmente, o módulo SiSeM está pronto, contém as metodologias XP, XP1 e RUP, e podem ser adicionadas outras futuramente. Com relação ao módulo ERA, estão prontos os sub-módulos: gráfico, armazenamento e recuperação da rede, de recursos humanos e simulação, utilizados no estudo de caso do projeto SAD/CHESF. Será necessário posteriormente desenvolver o módulo de comunicação para comunicar à equipe e gerente os próximos *deadlines* de etapas de projeto, além de implementar as novas diretrizes obtidas a partir dos experimentos realizados para melhorar a usabilidade do sistema.

References

- “APSEE – Prosoft – Projeto de Ambiente de Desenvolvimento de Software”(2005) UFRGS, <http://www.inf.ufrgs.br/prosoft>. Março.
- Aversano, L.; Gaeta, M; DeLucia, A.; Ritrovato, P.; Stafanucci, S.; Villani, M.L. Managing Coordination and Cooperation in Distributed Software Process: the GENESIS Environment, (2005). Wiley InterScience. <http://www.interscience.wiley.com>. December.
- Barbetta, P. A.; Reis, M.M.; Bornia, A.C. (2008) Estatística para cursos de engenharia e Informática, 2ª Edição, São Paulo, Editora Atlas S.A.
- Chen, Y.L.; Hsu, P.Y.; Chang, Y.B. A Petri Net Approach to Support Resource Assignment in Project Management. IEEE Transactions on Systems, Man and Cybernetics Part A: Systems And Humans, Vol. 38, No. 3, May, 2008.
- Cukierman, Z. S. Modelo PERT/CPM Aplicado A Projetos (2001) ed. Reichmann E Autores.
- DotProject (2006) <http://www.dotproject.net/>. Acesso em janeiro, 2008.

- Farias, C.B.A. Gerenciamento de Projetos com Redes de Atividades, (2007) Qualificação de Doutorado, UFCG/CCEI/DEE/COPELE, maio.
- Hasegawa, K.; Miyagi, P. E.; Santos Filho, D. J. , Takahashi, K.L. Ma, and Sugisawa, M. “On resource arc for Petri net modeling of complex resource sharing system,” (1999) J. Int. Robot. Syst. Theory Appl., vol. 26, no. 3/4, pp. 423–437.
- Jacobson, I.; Booch, G.; Rumbaugh, J. (1999) The Unified Software Development Process, Addison-Wesley Publishing Company.
- Meredith, J. R; Mantel, S. J. (2000) Project Management: A Managerial Approach, 4th ed. New York: Wiley.
- “Minute Man”. (2003) <http://www.minuteman-systems.com> Outubro.
- “MS Project.” (2005) <http://www.microsoft.com/brasil/office/project/default.asp>. Outubro.
- Murata, T. (1989) “Petri Nets: Properties, Analysis and Applications”. Proceedings of the IEEE, 77(4): 541-580. April.
- “ODE - Um Ambiente de Desenvolvimento de Software Baseado em Ontologias” (2005) <http://www.inf.ufes.br/~labes/ode>. Outubro.
- Queiroz, J. E. R. (2001) Abordagem Híbrida para Avaliação de Usabilidade com Interfaces com o Usuário. Tese, UFCG, CEEI, DEE, COPELE, Campina Grande.
- “RUP BUILDER - Rational Unified Process Builder.” Process Manager’s Guide, (2004) <http://sdt.web.cern.ch/sdt/RUP/builderguide.pdf>. Janeiro.
- SAD – Sistema de Apoio à Decisão, CHESF/UFCG/DSC, (2007). http://www.chesf.gov.br/downloads/pesquisa%20e%20desenvolvimento/Projetos/PP_83.pdf Junho.
- “TABA. Meta-Ambiente para instanciação de Ambientes de Desenvolvimento convencionais e Orientados a Domínios (2004). <http://www.cos.ufrj.br/taba>. Janeiro.
- Torres, J. B. “Uma Ferramenta de Gerência de Projeto – GEPRO” (1996). Dissertação de Mestrado, COPIN - UFPB, Campina Grande.
- Vianna, M. Conheça o Microsoft Solutions Framework - MSF. (2004). http://www.linhadecodigo.com.br/artigos.asp?id_ac=78&sub=0. Março.
- “Visual Studio Team System – VSTS”. (2008) <http://www.microsoft.com/brasil/msdn/teamsystem/Default.mspix>. Fevereiro.
- Wang, C.; Xue, L.J.; Yang, L. Research of College's Scientific Research Management Workflow Model Based on Petri Net. School of Management (2007) International Conference on Management Science & Engineering (14th), 20-22 Harbin, P.R.China, August.
- “XP - eXtreme Programming” (2003) <http://www.extremeprogramming.org>. Setembro.
- “XPLANNER”. (2005) <http://www.xplanner.org>. Outubro.
- “XP1-Adaptação de eXtreme Programming”. (2007) <http://dsc.ufcg.edu.br/~jacques/projetos/common/xp1/xp1.html>. Março.