

Sistema de Gerenciamento de Workflow baseado em Redes de Petri e em Algoritmos Genéticos

Fernanda Francielle de O. Malaquias¹, Webert Vieira Arantes¹
Edgard Afonso Lamounier Jr.¹, Alexandre Cardoso¹

¹Programa de Pós-Graduação em Engenharia Elétrica, FEELT-UFU,
Universidade Federal de Uberlândia, Av. João Naves de Avila, 2160
Uberlândia-M.G.-Brasil

fernandafrancielle@yahoo.com.br, webert3d@yahoo.com.br
lamounier@ufu.br, alexandre@ufu.br

Abstract. *In order to face new challenges, organizations have been using technologies to deal with information in an efficient way. The use of Workflow Management Systems (WFMS) has been particularly important in this context. However, in order to be successful in the implementation of these systems, organizations must believe not only in the correct modelling of their business process but also in the use of optimization techniques for real time scheduling. From this perspective, this paper aims to propose a WFMS that is based on Petri nets for the formal modelling of the processes and based on Genetic Algorithms for the real time scheduling. The proposed technique was tested in a prototype with a 3D interface. Comparisons were made with the FIFO technique, and in all cases the Genetic Algorithm was more efficient.*

Resumo. *A fim de enfrentar novos desafios, as empresas têm adotado tecnologias que lhes permitem lidar com a informação de maneira eficiente. O uso de Sistemas de Gerenciamento de Workflow (SGW) tem sido importante nesse contexto. No entanto, para que haja sucesso na implementação destes sistemas, as empresas precisam apostar não só na modelagem adequada dos seus processos de negócio, mas também no uso de técnicas de otimização para o escalonamento de tarefas em tempo real. Diante disso, este artigo visa propor um SGW que, se baseia em redes de Petri para a modelagem formal de processos e em Algoritmos Genéticos (AG) para o escalonamento em tempo real. A técnica proposta foi testada em um protótipo com interface 3D. Foram realizadas comparações com a técnica FIFO e em todos os casos o AG se mostrou mais eficiente.*

1. Introdução

Em um contexto de grandes e rápidas mudanças, as organizações estão adotando novos modelos de gestão e de organização social do trabalho que sejam mais flexíveis, mais participativos e com estruturas hierárquicas mais reduzidas. A adoção destes novos modelos de trabalho é acompanhada pela utilização de tecnologias e sistemas de informação, tais como o correio eletrônico, sistemas de suporte à decisão, Sistemas de

Gerenciamento de *Workflow* (SGW) etc. Dentre os sistemas citados, destacam-se os SGW que se tornaram, nos últimos anos, um componente padrão para sistemas de informação empresariais.

Os SGW [Aalst e Hee 2002] são pacotes de software genéricos para gerenciamento de processos de negócio. Estes sistemas apresentam-se como uma solução capaz de melhorar a eficiência e a gestão dos processos organizacionais, uma vez que provêm uma automação do gerenciamento de processos de negócio e a alocação dos recursos apropriados associados com as diversas atividades. Esses sistemas permitem o gerenciamento das tarefas fazendo com que a informação certa chegue até a pessoa certa no momento certo.

Segundo o manual de referência da *Workflow Management Coalition* [Hollingsworth 1995], os SGW, de um modo geral, precisam suportar as seguintes funcionalidades:

- funções referentes à definição e modelagem dos processos de negócio e das atividades que constituem estes processos;
- funções de controle em tempo de execução, que gerenciam e executam os processos de *Workflow* em um ambiente operacional, além de sequenciar e alocar as atividades necessárias para a execução desses processos; estas funções são responsáveis pelo escalonamento das tarefas e pelo acompanhamento de cada caso até o término de sua execução;
- interações em tempo de execução entre usuários humanos e ferramentas de Tecnologia da Informação.

Este trabalho está focado nas duas primeiras funcionalidades.

No que se refere à função de definição e modelagem dos processos, muitas pesquisas, já foram realizadas [Aalst e Hee 2002], [Coves *et al.* 1998], [Li *et al.* 2003]. Cada autor utiliza uma técnica de modelagem diferente, isto é, não há um modelo conceitual comum, aceito e usado por todos na área de *workflow*. No entanto, é importante ressaltar que a maioria destas pesquisas considera mais a dimensão dos processos relacionada ao sequenciamento das atividades. Elas possuem algumas limitações no que se refere à representação de alocação de recursos e de restrições temporais. Em [Aalst e Hee 2002], por exemplo, o mecanismo de alocação de recursos é representado apenas de modo informal e em [Coves *et al.* 1998] e [Li *et al.*, 2003] as restrições de tempo nem são especificadas.

Quanto à função de escalonamento em tempo real dos SGW, o trabalho em [Aalst e Hee 2002] aponta a habilidade limitada dos SGW no que se refere à alocação dos recursos e à decisão com relação à ordem em que as tarefas que usam os mesmos recursos devem ser executadas. Segundo Tramontina *et al.* (2004) não há resultados na literatura sobre escalonamento que cubram uma porção significativa das características dos SGW.

A fim de solucionar o problema do escalonamento dos SGW, em [Oliveira 2004], [Oliveira e Julia 2006] e [Julia, Oliveira e Valette 2008] foi proposto um modelo baseado em redes de Petri que permite representar tanto recursos discretos, quanto recursos contínuos. Também foi utilizada uma técnica de propagação de restrições que

torna possível calcular os tempos que um caso pode esperar entre uma atividade e outra, para que o seu tempo de execução esteja dentro do intervalo permitido para o processo em consideração. A proposta se mostrou eficiente e o resultado da simulação foi dado por um cenário admissível correspondente a uma sequência específica de atividades que respeita as restrições temporais. A limitação dessa técnica é que podem ocorrer situações em que a borda máxima de um intervalo de visibilidade seja alcançada, não havendo mais garantia de que todos os prazos sejam respeitados.

Visando expandir os resultados obtidos em [Oliveira 2004], [Oliveira e Julia 2006] e [Julia, Oliveira e Valette 2008], a proposta deste artigo para solucionar o problema do escalonamento é a combinação da técnica baseada em Redes de Petri proposta pelos autores aliada a um algoritmo genético capaz de determinar uma sequência de atividades a serem realizadas por recursos compartilhados híbridos, de modo a minimizar o número de casos atrasados.

Na próxima seção será apresentado um referencial teórico sobre o problema do escalonamento dos SGW e nas demais seções será apresentada a abordagem proposta bem como o protótipo de SGW que foi implementado para realização de testes.

2. Problema do Escalonamento dos Sistemas de Gerenciamento de *Workflow*

O comportamento dinâmico dos SGW impõe um escalonamento do controle de fluxo. O problema de escalonamento [Esquirol e Lopez 1995] consiste em organizar no tempo a sequência das tarefas considerando restrições temporais (intervalos de tempo) e restrições de utilização dos recursos compartilhados necessários à execução das tarefas.

As restrições temporais descrevem as relações de ordem relativa entre as diversas tarefas e são, geralmente, derivadas de imperativos relacionados à gestão do processo, aos limites de duração das atividades ou à duração total do processo. Já as restrições de utilização dos recursos exprimem a natureza e a quantidade dos recursos utilizados para as tarefas assim como as características de utilização desses recursos. Uma solução para o problema do escalonamento de sistemas de gerenciamento de *workflow* deve considerar características importantes inerentes a estes sistemas.

A primeira delas se refere à natureza dos recursos utilizados. Nos sistemas de produção, os recursos representam equipamentos físicos e são representados por simples fichas nos lugares de uma Rede de Petri. Eles são recursos do tipo discreto. No caso dos SGW, recursos podem representar tanto equipamentos físicos quanto funcionários humanos. A maioria dos recursos humanos pode tratar diversas atividades simultaneamente e por isso, nem sempre recursos deste tipo podem ser representados por simples fichas nos lugares de uma rede de Petri.

Além disso, em SGW, muitos casos podem ser executados simultaneamente e situações de conflito, que devem ser resolvidas em tempo real, podem ocorrer se um mesmo recurso não preemptivo for chamado ao mesmo tempo para a execução de atividades que pertencem a diferentes cenários.

Outras características que devem ser consideradas são as restrições temporais relativas aos prazos de entrega para os casos em um processo além dos diferentes roteiros que cada caso pode seguir. Vários roteiros são possíveis em um Processo de Negócio: há o roteiro sequencial (as atividades são executadas uma após a outra) o

roteiro paralelo (duas ou mais atividades realizadas em paralelo) e o roteiro alternativo (há uma escolha entre duas ou mais alternativas).

Uma abordagem clássica para solucionar o problema de escalonamento de sistemas representados por grafos com restrições disjuntivas [Esquirol e Lopez 1995] é combinar um mecanismo de propagação de restrições que permite o cálculo de intervalos de datas admissíveis e um algoritmo *branch and bound* cuja meta é encontrar uma sequência ótima das tarefas que respeite as restrições de tempo.

Esta abordagem apresenta uma limitação no caso de SGW, pois a solução é dada, geralmente, por uma sequência perfeita das atividades que será difícil de seguir em tempo real para sistemas que consideram o comportamento humano.

A maioria dos SGW atuais processam os casos na ordem em que eles chegam no sistema, ou seja, baseados em uma política FIFO (*First In First Out*), onde o primeiro a chegar é o primeiro a sair. Esta técnica também apresenta suas limitações e geralmente não funciona como a melhor estratégia [Tramontina *et al.* 2004].

2.1. Algoritmos Genéticos aplicados ao problema do escalonamento dos SGW

Os Algoritmos Genéticos (AG) [Goldberg 1989] são algoritmos de busca que se baseiam em mecanismos de seleção natural. Combinam sobrevivência dos mais aptos e cruzamento aleatório de informação.

Os componentes essenciais dos algoritmos genéticos são:

- Uma população de tamanho constante inicializada randomicamente, sendo que cada indivíduo desta população representa um ponto no espaço de busca para um dado problema;
- um valor de aptidão que é atribuído a cada indivíduo da população;
- classificação e seleção de indivíduos de acordo com a sua aptidão de forma que os mais aptos tenham mais chance de se reproduzir.

Operações genéticas como *crossover* (cruzamento) e mutação são aplicadas a indivíduos ou a pares de indivíduos a fim de produzir novos indivíduos, isto é, novas possíveis soluções para o problema.

Cada iteração do algoritmo é chamada de uma geração, já que uma nova população derivada da antiga é criada. A condição de parada pode ser um número máximo de gerações para o algoritmo, ou quando não é mais possível encontrar um indivíduo com maior aptidão, ou ainda uma combinação dessas duas. O retorno dado pelo algoritmo genético é o melhor indivíduo por ele encontrado, ou seja, a sua melhor resposta para o problema.

Os algoritmos genéticos têm sido utilizados para solucionar diferentes problemas de otimização, dentre eles problemas de escalonamento. Com o objetivo de auxiliar na definição ou otimização de modelos de processo, em [Guimaraes 2007], por exemplo, foi apresentada uma proposta baseada em algoritmos genéticos para *Flexible Job Shop Problems* (FJSP) e a integração do escalonamento FJSP a *Workflow* através de uma arquitetura para a modelagem de processos denominada *Workflow* Genético.

Visando diminuir o número de casos atrasados e minimizar o tempo de processamento dos casos, em [Tramontina *et al.* 2004] foram utilizadas técnicas de escalonamento em um ambiente dinâmico de *Workflow* e foram avaliados os desempenhos dessas técnicas frente à regra FIFO. Uma técnica denominada “*guess and solve*” foi proposta. Esta técnica consiste em prever os tempos de execução e rotas das atividades e resolver o problema de escalonamento determinístico resultante com técnicas como as regras de prioridade e algoritmos genéticos.

A limitação destas pesquisas é que a solução apresentada para o controle em tempo de execução procura decompor um problema de escalonamento dinâmico em uma série de problemas estáticos.

Um escalonamento estático é aquele em que o conjunto das informações necessárias à sua resolução é fixo (conjunto de tarefas, de recursos e suas características). A solução é um plano previsional e possui, portanto, uma validade com duração limitada devido ao aparecimento de novas informações que ocorrem durante a execução do plano, como por exemplo o aparecimento de novos casos, roteiros alternativos (não é possível prever qual rota cada caso irá seguir), atrasos e outros. Sendo assim, um controle dinâmico da execução é necessário; este controle deve ser constituído de uma função de decisão em tempo real que realizará os ajustes necessários e como consequência irá satisfazer os objetivos globais do plano.

3. Modelagem de Sistemas de Gerenciamento de *Workflow*

Para ilustrar a abordagem proposta por este artigo o processo de um “Serviço de Reclamações” apresentado em [Oliveira 2004] será utilizado. Neste processo, primeiro uma reclamação é registrada. Depois, o cliente que fez a reclamação e o departamento afetado pela reclamação são contactados. O primeiro é contactado a fim de se obter mais informações. O segundo é informado sobre a reclamação e é questionado sobre uma reação inicial. Essas duas atividades devem ser executadas em paralelo. Depois disso, as informações são coletadas e uma decisão é tomada. Dependendo da decisão, ou um pagamento é feito ou uma carta é enviada ao cliente. Finalmente, a reclamação é arquivada.

3.1. Modelagem baseada em redes de Petri híbridas p-temporais

3.1.1. Definição de Roteiros

Baseando-se nos requisitos iniciais de um Sistema de Gerenciamento de *Workflow*, é possível obter um modelo de rede de Petri correspondente, representando cada atividade por um lugar específico, com uma transição de entrada que mostra o início da atividade e uma transição de saída que mostra o final da atividade.

A Figura 1 mostra a rede de Petri para o “Serviço de Reclamações”, onde as atividades são representadas pelos lugares A_i (para $i=1$ até 8) e os tempos de espera entre uma atividade e outra são representados pelos lugares E_j (para $j=0$ até 9). Os lugares de espera “E3” e “E4” são necessários para sincronizar o final das atividades “Contactar Cliente” e “Contactar Departamento”.

3.2. Alocação de Recursos

Como mencionado anteriormente, os recursos em sistemas de *workflow* podem ser tanto equipamentos quanto recursos humanos. Alguns desses recursos são do tipo discreto e podem ser representados por simples fichas em uma rede de Petri. Por exemplo, uma impressora usada pra tratar uma classe específica de documentos será representada como um recurso não preemptivo e pode ser alocada para um único documento em um mesmo momento. Mas, existem alguns recursos que não podem ser representados por uma simples ficha. Isso é o caso da maioria dos recursos do tipo humano.

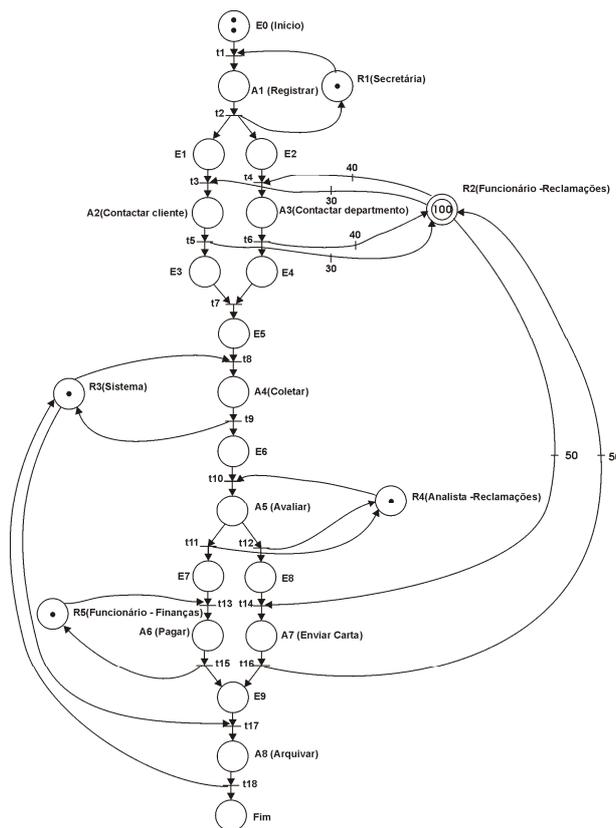


Figura 1. Modelo de rede de Petri com alocação de recursos híbridos para o “Serviço de Reclamações”

De fato, não é incomum para um funcionário que trabalha em um escritório tratar muitos casos simultaneamente. Por exemplo, em uma companhia de seguros, um funcionário pode normalmente tratar vários casos durante um dia de trabalho e não necessariamente de uma maneira puramente sequencial. Neste caso, uma simples ficha discreta não poderia representar o comportamento humano de uma maneira apropriada.

Como solução para este problema, alguns dos recursos de tipo humano serão representados por números reais (iguais a cem quando o recurso estiver 100% disponível) que mostrará a disponibilidade do recurso.

Os mecanismos de alocação de recursos pra o “Serviço de Reclamações” estão representados na Figura 1. A “Secretária”, o “Funcionário – Reclamações”, o

“Funcionário – Finanças” e o “Sistema” são representados como recursos discretos. Estes recursos tratarão as atividades em uma ordem puramente sequencial. Por exemplo, a “Secretária” representada por uma ficha em R1 pode somente registrar os dados de um cliente de uma maneira não preemptiva. Ao contrário, o funcionário do departamento de reclamações é representado pelo número 100 em R2. Este recurso contínuo pode ser usado para tratar as atividades “Contactar Cliente”, “Contactar Departamento”, “Enviar Carta”. A atividade “Contactar Cliente” irá precisar de 35% da disponibilidade desse funcionário, a atividade “Contactar Departamento” irá precisar de 40% e, finalmente, a atividade “Enviar Carta” irá precisar de 50% de sua disponibilidade. Isso significa que, de certa maneira, o funcionário representado no lugar R2 poderá processar mais de uma atividade em um dado espaço de tempo. Por exemplo, ele poderá escrever uma carta (50% da disponibilidade do recurso R2 para a atividade “Enviar Carta”) e, ao mesmo tempo, tentar ligar para um cliente (35% da disponibilidade do recurso R2 para a atividade “Contactar Cliente”). Estes valores foram definidos apenas a título de exemplificação, não sendo portanto valores de empresas reais.

3.3. Modelo Temporal

Como o tempo real necessário para a execução de uma atividade em um sistema de *workflow* é não-determinístico e não é facilmente previsível, um intervalo de tempo pode ser atribuído a cada atividade a fim de representar um tempo mínimo e um tempo máximo para a duração da atividade.

O modelo p-temporal para o “Serviço de Reclamações” é mostrado na Figura 2. Por exemplo, o intervalo estático $[20, 30]_e$ associado ao lugar A_2 que representa a atividade Contactar Cliente, significa que o Funcionário do Departamento de Reclamações (Recurso R2) que executará essa atividade precisará de no mínimo 20 unidades de tempo e no máximo 30 unidades de tempo. Os intervalos estáticos associados às atividades “Coletar Informações” em A_4 e “Arquivar” em A_8 são iguais a $[0, 0]$ porque a duração dessas atividades é desprezível se comparada com a duração das outras atividades do “Serviço de Reclamações”.

Os tempos de espera existentes entre as atividades podem ser representados por um intervalo de visibilidade (intervalo de datas) cujas bordas mínimas e máximas irão depender das datas mínimas e máximas de entrega para o caso considerado.

Por exemplo, na Figura 3, se a data de chegada de uma ficha no lugar P é $\delta = 3$, sabendo-se que o intervalo estático neste lugar é $[5, 9]_e$, então, o intervalo de visibilidade desta ficha é $[5 + 3, 9 + 3]_v = [8, 12]_v$, ou seja, a ficha deverá permanecer no lugar P no mínimo até a data 8 e no máximo até a data 12.

Conhecendo a data de início e a máxima duração de um caso, é possível calcular os intervalos de visibilidade associados às fichas nos lugares de espera usando técnicas de propagação de restrições propostas em [Julia, Oliveira e Valette 2008].

Aplicando tais técnicas de propagação de restrições ao exemplo do “Serviço de Reclamações”, o resultado é mostrado na Figura 2. É considerado que a máxima duração permitida para um caso do “Serviço de Reclamações” é 105 unidades de tempo.

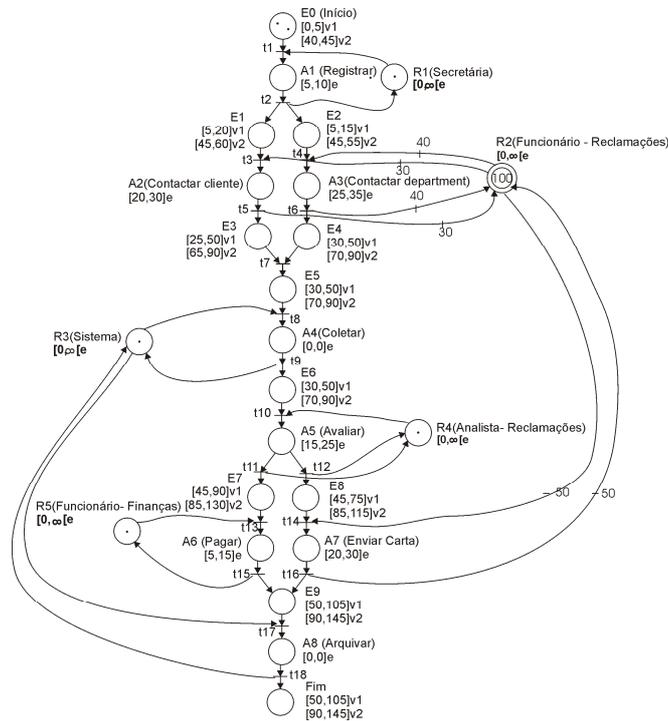


Figura 2. Modelo p-temporal para o “Serviço de Reclamações” após aplicação dos mecanismos de propagação de restrição

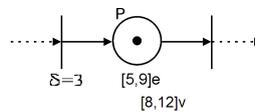


Figura 3. Intervalo de Visibilidade

Na rede de Petri da Figura 2 dois casos, representados por duas fichas no lugar E0, são considerados. O primeiro caso pode ser iniciado na data 0 que é representada como a borda mínima do intervalo de visibilidade v1 em E0. O segundo caso pode ser iniciado na data 40 que é representada como a borda mínima do intervalo de visibilidade v2 em E0. Como a máxima duração permitida para cada caso é de 105 unidades de tempo, a borda máxima do intervalo de visibilidade associado ao último lugar “Final” do processo será igual a 0 (data de início do caso 1) + 105 (máxima duração de um caso) = 105 (máxima data para a entrega do caso 1) para o primeiro caso e 40 (data de início do caso 2) + 105 (máxima duração de um caso) = 145 (máxima data para a entrega do caso 2) para o segundo caso. As bordas dos outros intervalos de visibilidade associados aos casos nos lugares de espera foram calculadas aplicando-se as técnicas de propagação de restrição aos diferentes roteiros do “Serviço de Reclamações”.

4. Escalonamento em tempo real baseado em Algoritmos Genéticos

Para resolver um problema de escalonamento, pode-se optar por obter uma solução ótima ou uma solução admissível. A abordagem por otimização supõe que as soluções candidatas a um problema possam ser ordenadas de maneira racional segundo um ou mais critérios de avaliação numérica que são vistos como indicadores de desempenho. Procura-se, então, minimizar ou maximizar tais critérios. Os critérios podem ser

relacionados ao tempo (tempo total de execução ou tempo médio de finalização de um conjunto de tarefas, quantidade de casos atrasados etc.), aos recursos (quantidade de recursos necessários para realizar um conjunto de tarefas), aos custos de produção etc.

O algoritmo genético implementado neste trabalho teve como objetivo determinar uma sequência de atividades a serem realizadas por recursos compartilhados híbridos, de modo a minimizar o número de casos atrasados. Isso porque evidências encontradas em [Bierwirth e Mattfeld 1999], [Jain e Meeran 1998] apontam para o problema de minimizar o número de casos atrasados como mais factível de resolução por AGs; além disso, minimizar o tempo de processamento não é uma boa estratégia para sistemas que devem considerar o comportamento humano [Oliveira e Julia 2006].

Os indivíduos foram representados por sequências de atividades. Cada indivíduo é dado por uma permutação de todas as atividades $A(i,j)$ de todos os casos em espera por um recurso compartilhado em um dado instante.

Considerando-se, por exemplo, os dois casos (caso 1, caso 2) representados na rede de Petri da Figura 2, ambos serão processados pelo recurso R2. Portanto, as atividades $A_{2,1}$ e $A_{3,1}$, $A_{2,2}$ e $A_{3,2}$ (atividades A_2 e A_3 do primeiro e segundo caso respectivamente) e, possivelmente, $A_{7,1}$, $A_{7,2}$ (atividade A_7 do primeiro caso e segundo caso respectivamente) serão realizadas pelo recurso R2 dependendo do roteiro seguido por cada caso.

Supondo que em um dado instante, as atividades $A_{2,2}$, $A_{3,2}$ e $A_{7,1}$ estejam em estado de espera pelo recurso R2. Qualquer permutação destas pode ser considerada uma representação de uma solução para o escalonamento.

A população inicial do AG é gerada por um processo que parte de uma primeira permutação p_0 das atividades colhidas junto ao sistema. Uma segunda permutação é gerada escolhendo-se aleatoriamente duas atividades de p_0 e trocando suas posições. Neste momento a população inicial possui dois elementos válidos. Em seguida escolhe-se aleatoriamente um deles para servir de origem a uma nova representação pelo processo já mencionado e assim por diante até obter o número de indivíduos total da população inicial.

O operador de *crossover* utilizado foi o PMX (*Partially Matched Crossover*). Como muitos outros operadores, o PMX seleciona duas representações pais e gera uma nova representação filha a partir delas. Para a probabilidade de cruzamento foi utilizado o valor de 0.6 (60%). O processo de mutação consistiu em realizar a troca de posições de duas atividades de um indivíduo. Novamente escolhem-se estas atividades aleatoriamente. Para a probabilidade de mutação foi utilizado o valor 0.1 (10%).

A função de aptidão utilizada para a avaliação da aptidão dos indivíduos foi: $\text{fitness}(I_k) = \text{nlate}(I_k)$, sendo $\text{nlate}(I_k)$ o número de casos atrasados segundo a sequência de I_k .

O valor de $\text{nlate}(I_k)$ é calculado considerando o maior instante no qual uma atividade $A(i,j)$ tem possibilidade de começar sua execução. Esse instante é o menor entre os tempos máximos de duração das atividades que precedem $A(i,j)$ na ordem cronológica do caso j . Se esse instante pertencer ao intervalo de visibilidade de espera do caso j , isso significa que o caso não atrasará, caso contrário soma-se 1 ao $\text{nlate}(I_k)$ indicando que um atraso ocorrerá se a sequência representada por I_k for seguida.

Por exemplo, no caso do recurso R2 que pode executar mais de uma atividade simultaneamente. Caso, as atividades $A_{2,2}$, $A_{3,2}$ e $A_{7,1}$ estejam alocadas nesta sequência, as atividades $A_{2,2}$, $A_{3,2}$ serão executadas paralelamente e a atividade $A_{7,1}$ só será iniciada quando uma das atividades $A_{2,2}$ ou $A_{3,2}$ terminar. Portanto, para verificar se haverá atrasos ou não diante desta sequência de execução, é considerado o tempo que ainda falta para que uma das duas atividades termine. Se o instante de término da atividade que possivelmente terminará primeiro pertencer ao intervalo de visibilidade da espera para execução de $A_{7,1}$ (E8), então não haverá atraso. Caso contrário, um atraso poderá ocorrer e será contabilizado.

5. Protótipo de SGW com visualização 3D

Para testar a abordagem proposta foi desenvolvido um protótipo de um SGW que contém dois módulos, o primeiro referente à definição e modelagem dos processos de negócio e o segundo que contém funções de controle em tempo de execução.

O seu desenvolvimento foi baseado na arquitetura de software proposta em [Oliveira e Julia 2006]. A Figura 4 mostra a tela usada para modelagem das atividades, a tela de modelagem dos recursos e a tela para a configuração dos parâmetros do algoritmo genético.

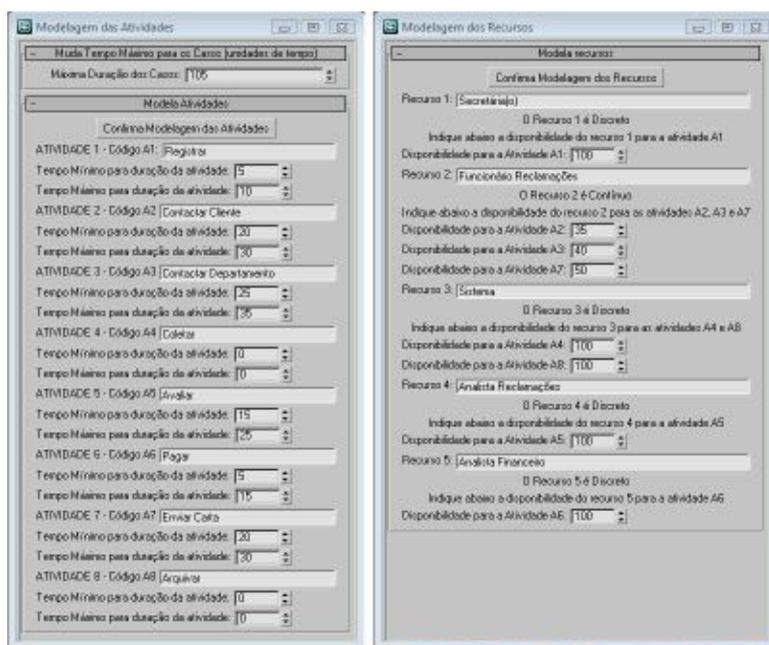


Figura 4. Telas de definição das atividades e dos recursos

A Figura 5 mostra a maquete 3D criada especificamente para o serviço de reclamações. Esta maquete foi utilizada para simulação em tempo real do algoritmo proposto e possibilitou uma melhor visualização dos resultados obtidos. Por meio dela, é possível visualizar em tempo de execução quantos e quais casos (representados por esferas) estão alocados a cada recurso. Esferas de cores diferentes foram utilizadas para representar atividades em execução (azul), casos em espera (amarelo) e casos em atraso (vermelho). Ao passar o mouse sobre as esferas os dados sobre o caso correspondente são apresentados.

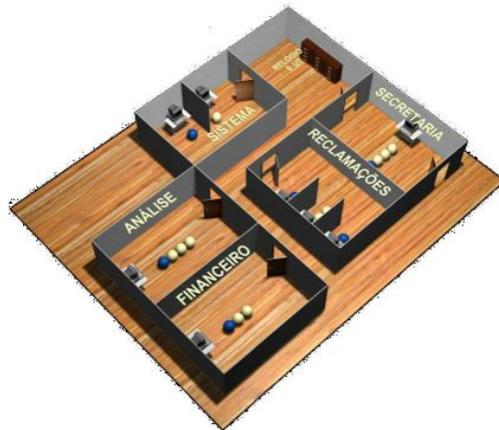


Figura 5. Maquete para Simulação do “Serviço de Reclamações”

Entende-se que este tipo de interface possa ser utilizada como ferramenta de apoio à decisão por gerentes que poderão encontrar gargalos tais como as sobrecargas de recursos de forma visual e intuitiva.

Várias simulações foram realizadas para comparar a técnica utilizada com a técnica FIFO. Em cada simulação foram utilizados a mesma quantidade de casos disparados com o mesmo intervalo de tempo. Em todos os casos o Algoritmo Genético se mostrou eficiente, reduzindo o número de casos atrasados em pelo menos 50%.

5. Considerações Finais

Este artigo apresentou a proposta de um Sistema de Gerenciamento de *Workflow* que permite a especificação inicial de um processo de negócio com explícitas considerações de tempo e mecanismos de alocação de recursos híbridos. Baseado neste modelo, um Algoritmo Genético usado para o problema de escalonamento foi aplicado. O resultado é dado por uma sequência de atividades que procura respeitar as restrições de tempo e de alocação dos recursos, minimizando os atrasos ao longo dos processos.

As vantagens de utilização da abordagem proposta são diversas. Primeiramente o mecanismo de tomada de decisão em tempo real baseado em algoritmos genéticos permite gerar sequências de atividades com base nos tempos de espera calculados, reduzindo a possibilidade de ocorrer atrasos.

Além disso, a solução proposta considera os diferentes tipos de recursos utilizados, o compartilhamento de recursos, as restrições temporais relativas à duração de cada atividade e aos prazos de entrega para os casos além dos diferentes roteiros que cada caso pode seguir seja ele sequencial, paralelo ou alternativo. Tudo isso é considerado em tempo de execução, deixando de ser necessário prever o comportamento dos casos para depois escalonar, o que mostra o caráter inédito da solução proposta.

Outra contribuição deste trabalho está na proposta de um novo tipo de interface 3D que permite aos gestores visualizar em tempo real o fluxo de informações ao longo do processo facilitando assim o processo de tomada de decisão.

Como trabalho futuro, pretende-se desenvolver um módulo que contemple a terceira funcionalidade dos SGW (citada na introdução desse trabalho) e integrá-lo aos outros dois módulos já desenvolvidos.

Referências

- Aalst, W. van der; Hee, K. van (2002). *Workflow Management: Models, Methods and Systems*. The MIT Press. Cambridge, Massachusetts. 368p.
- Bierwirth, C; Mattfeld, D. C. (1999). Production scheduling and rescheduling with genetic algorithms. *Evolutionary Computation*, 7:1-17, 1999.
- Coves, C.; Crestani, D.; Prunet, F. (1998). “Design and Analysis of Workflow Processes with Petri nets”. In: 1998 IEEE International Conference on Systems, Man, and Cybernetics Proceedings of IEEESMC’1998. vol. 1. p. 101-106.
- Esquirol, P., Huguet, M.J., Lopez, P. (1995). Modelling and managing disjunctions in scheduling problems. *Journal of Intelligent Manufacturing* 6. pp. 133-144.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley. 412p.
- Guimarães, K. F.; (2007) Escalonamento Genético FJSP com tempo de configuração dependente de sequência. Dissertação. Universidade Federal de Uberlândia.
- Hollingsworth, D. (1995). “The Workflow Reference Model. Workflow Management Coalition Document Number TC00-1003”. Document Status - Issue 1.1
- Jain, A. S., Meeran, S.; (1998) “A state-of-the-art review of job-shop scheduling techniques”. Technical report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland.
- Julia, S.; de Oliveira, F. F.; Valette R. (2008). Real Time Scheduling of Workflow Management Systems Based on a p-time Petri net model with hybrid resources. *Simulation Modelling Practice and Theory*. Elsevier Science. p.462-482.
- Li, J.; Fan, Y.; Zhou, M. (2003). Timing Constraint Workflow Nets for Workflow Analysis. *IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans*. Vol. 33, N. 2, March 03.
- Oliveira, F. F. de; (2004). Escalonamento em tempo real dos Sistemas de Gerenciamento de Workflow baseado em um modelo de Rede de Petri híbrida p-temporal. Dissertação. Universidade Federal de Uberlândia.
- Oliveira, F. F. de; Julia, S.; (2006). “Software Architecture for the Real Time Scheduling of Workflow Management Systems based on a Petri net model”. In: Anais do SBES2006 - XX Simpósio Brasileiro de Engenharia de Software p. 1-15.
- Tramontina, G. B.; Wainer, J.; Ellis, C. (2004). “Applying Scheduling Techniques to Minimize the Number of Late Jobs in Workflow Systems”. In: 2004 ACM Symposium on Applied Computing. p. 1397-1403.