

Uma Abordagem de Integração de Simulação Baseada em Agentes e Mineração de Processos

Fernando Szimanski¹, Célia G. Ralha¹, Ricardo P. Jacobi¹

¹Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Caixa Postal 4466 – 70.919-970 – Brasília – DF – Brasil

fszimanski@gmail.com, {ghedini,rjacobi}@cic.unb.br

Abstract. *The agent-based software simulation is able to promote the specification and the process behavior verification, having been applied in several areas such as Biology, Engineering, Economics and Social Sciences. On the other hand, mining process provides the discovery of processes through automated systems event logs. In this paper, we propose a study integration of these two different approaches, which present an interesting potential for being complementary and allowing the improvement of processes in the organizational setting. At first, the two approaches can be properly integrated with focus on event handling. One possible way to of integration is to model the process preliminary and submit it with the simulation agents, and the events generated may be used to redefine the process model using several mining techniques. Through the suggested integration it will be possible to verify the existence of convergence in the process of defining and redefining organizational business processes.*

Resumo. *A simulação baseada em agentes de software é capaz de promover a especificação e verificação do comportamento de processos, tendo sido aplicada nas mais diversas áreas, como Biologia, Engenharias, Economia e Sociologia. Por outro lado, a mineração de processos proporciona a descoberta dos processos através de logs de eventos de sistemas automatizados. Neste trabalho, é proposto um estudo de integração destas duas diferentes abordagens, as quais apresentam um potencial interessante por serem complementares e permitirem a melhoria de processos no cenário organizacional. A princípio, as duas abordagens podem ser adequadamente integradas com foco no tratamento de eventos. Uma possibilidade de integração é modelar preliminarmente o processo e submetê-lo a simulação com agentes, sendo que os eventos gerados poderão ser utilizados para redefinir o modelo de processo utilizando diversas técnicas de mineração. Através de modelos de integração das duas abordagens será possível verificar a existência de convergência no processo de definição e redescoberta de processos de negócio das organizações.*

1. Introdução

Como é o caso de diversas empresas na sociedade moderna, olhar para os processos nas organizações globalizadas é importante para facilitar a busca dos objetivos principais da organização de forma mais eficiente, proporcionar resposta mais rápida aos clientes, promover a utilização mais eficaz dos recursos e melhorar a qualidade do trabalho [Georgakopoulos et al. 1995], [Eriksson and Penker 2000], [Laguna and Marklund 2004].

Neste sentido, a análise cuidadosa da descrição de processos pode ajudar o executor de cada atividade a conhecer melhor seu trabalho e proporcionar a interação com outro executor de forma mais eficaz. Gestores podem utilizar a descrição dos processos para identificar gargalos, ineficiências, defeitos no processo e problemas na coordenação. Enfrentar estes problemas melhorando os processos que foram definidos, pode levar a organização a alcançar seus objetivos e também formar uma base para automações em potencial e troca de *e-Business* por meio de sistemas de TI.

A literatura pertinente a esta temática aponta como definições clássicas que processos de negócio são uma estrutura, conjunto de atividades medidas especificamente para produzir uma saída com valor para o cliente ou mercado. Isso implica em uma forte ênfase sobre a forma como o trabalho é realizado dentro de uma organização, em contraste a ênfase ao foco no produto, ou seja, um processo é uma determinada ordem de trabalhos durante um tempo e espaço, com um início e fim, e definição clara de entradas e saídas [Davenport 1993],[Hammer and Champy 2003].

Levando em consideração que uma das melhores formas de gerar mais lucros para uma empresa é reduzir os custos do negócio, empresas vem gastando milhões na construção de processos de negócio complexos para infra-estrutura já existentes através da integração de aplicações heterogêneas. Neste sentido, a *Business Process Modeling Initiative* (BPMI) desenvolveu a norma padrão *Business Process Modeling Notation* (BPMN) [OMG 2008]. A BPMN tem como principal objetivo fornecer uma notação que seja facilmente compreensível tanto por usuários do negócio como para analistas, que criam modelos iniciais dos processos, desenvolvedores técnicos que através do uso da tecnologia são responsáveis por executar os processos e, finalmente, para pessoas de negócio que irão gerir esses processos.

A simulação e o processamento de eventos são cada vez mais importantes na área de *Business Process Modeling* (BPM), comprovados pela crescente atenção em *complex event processing* (CEP) conforme citado em [Barros et al. 2007], [Eckert and Bry 2009].

Baseado neste contexto, este artigo apresenta um estudo de integração de duas diferentes abordagens: simulação baseada em agentes e mineração de processos; as quais apresentam um potencial interessante por terem características complementares e permitirem a melhoria de processos no cenário organizacional com foco no tratamento de eventos.

O restante deste artigo está organizado da seguinte forma: a Seção 2 fornece uma visão geral envolvendo processos de negócio com simulação baseada em agentes discutindo o *Framework Agent Object Relationship* (AOR), um ambiente de simulação baseado em agentes capaz de gerar eventos de processos; a Seção 3 apresenta uma visão geral sobre mineração de processos; na Seção 4 é apresentada abordagens de de integração dos modelos utilizando as literaturas relacionadas; e na Seção 5 apresentamos as conclusões e trabalhos futuros.

2. Simulação Baseada em Agentes

Visto que a BPMN provê o suporte para projetar um processo de negócio, mas não proporciona um processo de melhoria no modelo da aplicação, pois não fornece uma semântica formal de comportamento, que é importante na especificação e verificação do comportamento dos processos. Na literatura encontramos abordagens tradicionais para tentar

resolver este problema: redes de petri, cadeia de processos dirigida por eventos, redes de fluxo de trabalho (*Workflow*) e a abordagem de simulação baseada em agentes (*Agent-Based Simulation - ABS*) [Pascalau et al. 2009].

Considerando a abordagem de agentes, um agente de software é capaz de assumir uma ação independente em nome de seu usuário ou proprietário [Wooldridge 2009]. Neste contexto, pelo menos semanticamente, um sistema de software genérico, na forma de um sistema de gestão de processos de negócios pode ser de fato um Sistema Multi-Agente (SMA).

Segundo [Wagner 2003], a ABS é um novo paradigma que tem sido aplicado em problemas de simulação para diversas áreas: Ciências Biológicas, Engenharias, Ciências Econômicas e Ciências Sociais. Em ABS um cenário de agentes que interagem uns com os outros e com seu ambiente, está sendo modelado e simulado, como um SMA. Os agentes participantes (seres humanos, instituições sociais, sistemas de software ou máquinas) podem executar ações, perceber seu ambiente e reagir a mudanças nele. Eles também têm um estado mental que inclui componentes como conhecimento / opiniões, objetivos, lembranças e compromissos. Comparado com os métodos tradicionais de simulação (equações matemáticas, simulação de eventos discretos, autômatos celulares e teoria de jogos), a ABS é menos abstrata e mais próxima a realidade.

O cerne desta seção está baseado na literatura apresentada nos trabalhos [Wagner 2003], [Pascalau et al. 2009], [Wagner 2009], [Wagner and Diaconescu 2009] e [Diaconescu and Wagner 2009]. A proposta destes trabalhos é um *framework* para simulação de processos baseada em agentes denominado *Agent-Object-Relationship (AOR)*, o qual utiliza mapeamento de modelos utilizando a notação BPMN. O AOR será apresentado na Seção 2.1.

2.1. Agent-Object-Relationship (AOR)

O trabalho [Wagner 2003] propõe o *framework* de simulação AOR, que se destina a ser usado como um ambiente de simulação de processos para ciência, engenharia, educação e entretenimento. A maior parte do *framework* é *open-source*, licenciado sob a *General Public License (GPL)*, sendo baseado em *XML AOR Simulation Language (AORSL)*, que é usado para expressar em alto nível declarativo modelos de simulação de cenários, independentemente da linguagem de programação. Conforme citado pelo autor, com o uso de AORSL, o AOR é o primeiro *framework* de simulação baseado em agentes cognitivos.

O simulador AOR pode ser implementado em diferentes linguagens de programação, mas deve estar em conformidade com a arquitetura e modelo de execução do AOR, conforme foi definido em [Wagner 2003]. O AORJavaSim, é um programa *standalone desktop*, com interface gráfica baseada em Java com visualização em 2D que permite observar simulação de execução quando visões adequadas foram definidas.

Um arquivo de cenário de simulação é expresso na linguagem de simulação AORSL e, em seguida, primeiro traduzido para código fonte em Java antes de ser compilado para Java Byte Code e executado, conforme indicado na Figura 1.

Um modelo de simulação consiste em um modelo de espaço opcional, conjunto de tipos de entidades e conjunto de regras de ambiente. Já um tipo de agente é definido por

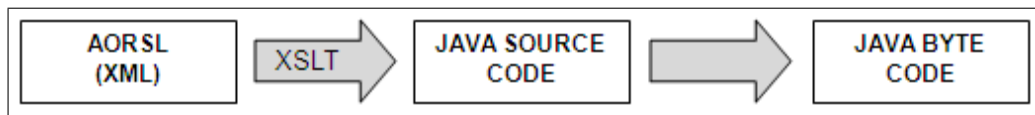


Figura 1. From AORSL to Java byte code [Wagner and Diaconescu 2009].

meio de um conjunto objetivo de propriedades, um conjunto subjetivo de propriedades, um conjunto subjetivo de tipos de entidade e um conjunto de regras do agente, que define o comportamento reativo do agente em resposta aos eventos.

O alto nível de categorias ontológicas de simulação do AOR são mensagens, eventos e objetos, que incluem os agentes, os objetos físicos e agentes físicos, como mostrado na Figura 2.

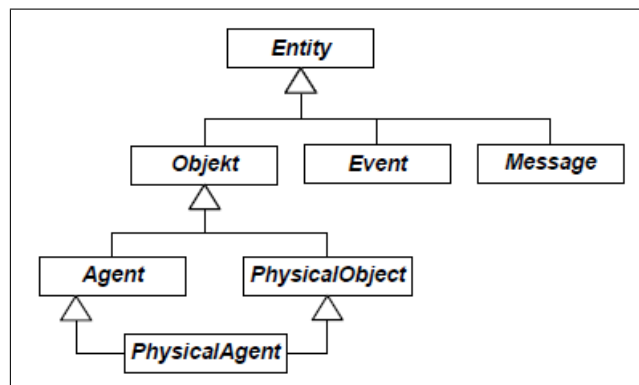


Figura 2. Upper-level ontological categories [Wagner and Diaconescu 2009].

Tanto o comportamento do ambiente como o dos agentes são modelados com a ajuda de regras, apoiando assim a modelagem do comportamento de alto nível declarativo.

O ambiente de regras é um conjunto de variáveis do tipo *5-tuple* (*EvtT*, *Var*, *Cond*, *UpdExpr*, *ResEvtExpr*), onde:

- *EvtT*: denota o tipo de evento que dispara a regra;
- *Var*: é um conjunto de declarações de variáveis;
- *Cond*: é uma fórmula de condição lógica para variáveis;
- *UpdExpr*: especifica uma atualização do estado do ambiente;
- *ResEvtExpr*: é uma lista de resultado de eventos.

O AOR tem o suporte para a distinção entre fatos e opiniões, incluindo a auto-opinião (opinião do agente sobre si mesmo). A linguagem de consulta baseada em SPARQL [Diaconescu and Wagner 2009] é usada para expressar consultas que um agente pode solicitar a outro agente sobre suas opiniões.

Portanto, a arquitetura do simulador AOR consiste em: um simulador de ambiente, que cria a percepção de eventos e passa aos simuladores de agentes interessados; um conjunto possivelmente vazio de simuladores de agentes, que processam quaisquer percepções dos eventos recebidos e, em resposta, criam eventos de ação que são passados de volta para o simulador de ambiente.

Ao executar uma simulação, o AOR cria um *log* em formato XML como base para geração de relatórios e análise estatística. Utilizando o *log* de eventos e a *engine*

open source de gráficos OGRE 3D, pode-se visualizar uma estrutura de animação em 3D através da Web. Tais *logs* podem também servir de base para mineração de processos que utilizam estas informações dos eventos gerados para redescobrir modelos de processos de negócio.

3. Mineração de Processos

Segundo [van der Aalst et al. 2003], o objetivo da mineração de processos é redescobrir o modelo de processo a partir da execução do comportamento de instâncias de processo, assumindo que é possível registrar eventos das tarefas que são executadas e identificar a instância do processo que produziu cada evento .

O registro das tarefas executadas pode ser obtido através de *log* de eventos gerados a partir do fluxo de trabalho de aplicações, onde são registrados as seqüências de eventos executados. Esta função pode se tornar difícil devido ao suporte limitado de aplicações existentes, complicando a recuperação de dados neste formato [van der Aalst et al. 2003].

O conteúdo desta seção está baseado na literatura apresentada nos trabalhos [Veiga and Ferreira 2009], [Ferreira et al. 2007], [Walicki and Ferreira 2010] e [Ferreira and Gillblad 2009]. Os trabalhos descrevem técnicas de mineração em *log* de eventos produzidos em instâncias de processos para descoberta de padrões.

3.1. Processos Estruturados

Processos estruturados facilitam a identificação da tarefa que foi executada, pois as atividades contidas no processo são rotuladas para que os *logs* dos eventos gerados sejam registrados com seqüências de entradas rotuladas das instâncias do processo, conforme visto no processo de suporte técnico ilustrado na Figura 3

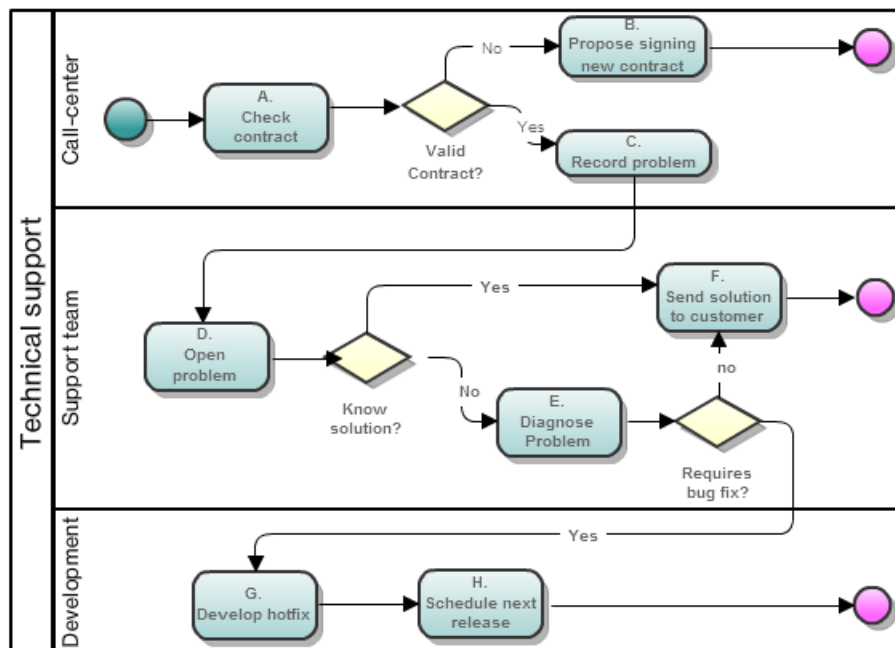


Figura 3. Processo de Suporte técnico [Ferreira and Gillblad 2009].

Segundo [Ferreira and Gillblad 2009] para facilitar a consulta, as atividades do fluxo de trabalho são rotuladas e na conclusão de cada atividade, um registro no *log* de

são rotulados, as técnicas de mineração de processos existentes geram modelos altamente complexos que muitas vezes são difíceis de entender.

Em aplicações reais, com eventos rotulados, também pode-se encontrar grande quantidade de ruído e variação de comportamento em *log* de eventos, estes são chamados de *spaghetti models*, tais variações tentou-se capturar com algoritmos de *clustering* no trabalho de [Veiga and Ferreira 2009].

Técnicas de clusterização tem sido investigadas como um meio para lidar com a complexidade de *spaghetti models*, dividindo os casos em *clusters*, levando a modelos menos confusos. Esta abordagem foi implementada no ProM [van Dongen et al. 2005], que é um *framework* extensível para mineração processos que inclui muitas técnicas para enfrentar os desafios nesta área.

Uma abordagem para mineração por clusterização já foi implementada no ProM, é conhecida como *Disjunctive Workflow Schema* (DWS). De acordo com esta metodologia, um *log* completo é examinado e um modelo é gerado utilizando *HeuristicsMiner*. Se o modelo gerado é o melhor e sem nenhum excesso de generalização a mineração é encerrada, caso contrário o registro é dividido em *clusters* utilizando o método de clusterização *K-means*.

Trace clustering [Song et al. 2008] é outra técnica implementada no ProM, que tem como objetivo particionar o *log* agrupando por seqüências semelhantes. Esta abordagem faz o uso de *distance-based clustering* com perfis, com a finalidade de reduzir a diversidade e a complexidade dos modelos através do baixo número de casos analisados ao mesmo tempo.

O campo da bioinformática tem sido a área onde se encontra a maioria das implementações de algoritmos de clusterização de seqüência, mas diversas pesquisas com este tipo de algoritmo têm sido realizada em outras áreas, como a análise de padrões de navegação em um site, onde estes padrões consistem na clusterização baseada nos caminhos de navegação em comum de usuários de um site. A clusterização de seqüências também foi usada no campo de mineração de processos, onde pode-se observar um experimento utilizando o *log* de eventos coletados manualmente nas atividades de uma equipe de desenvolvimento de software com o objetivo de descobrir a interação entre os membros da equipe; e também outro experimento realizado através da mineração de rotinas comuns em um sistema de banco de dados.

Embora o algoritmo de clusterização de seqüências que foi descrito anteriormente seja robusto a interferências, todas as seqüências devem ser atribuídas a um cluster. Caso uma seqüência seja muito incomum e diferente de todas as outras, ela irá afetar o modelo probabilístico dificultando ao final a interpretação do modelo do *cluster*. Para evitar este problema, o pré-processamento deve ser realizado para as seqüências de entrada antes da aplicação de clusterização de seqüências. O objetivo deste pré-processamento é limpar o conjunto de eventos eliminando assim seqüências indesejáveis, i.e, que raramente ocorrem. *Spaguetti models* são muitas vezes repletos de eventos obscuros que só contribuem negativamente para a descoberta do modelo de processo.

As etapas do pré-processamento são implementadas baseadas nas seguintes características: tipo do evento; eventos de apoio; repetições consecutivas; comprimento da seqüência; seqüências de apoio. Sendo que a ordem de aplicação destas etapas não podem

ser alteradas, senão pode-se obter resultados diferentes.

As etapas anteriores de pré-processamento e o algoritmo de *clustering* de seqüências foram implementadas e estão disponíveis como um novo *plug-in* para o framework de processos de mineração ProM. A Figura 5 ilustra as entradas e saídas para este *plug-in*.

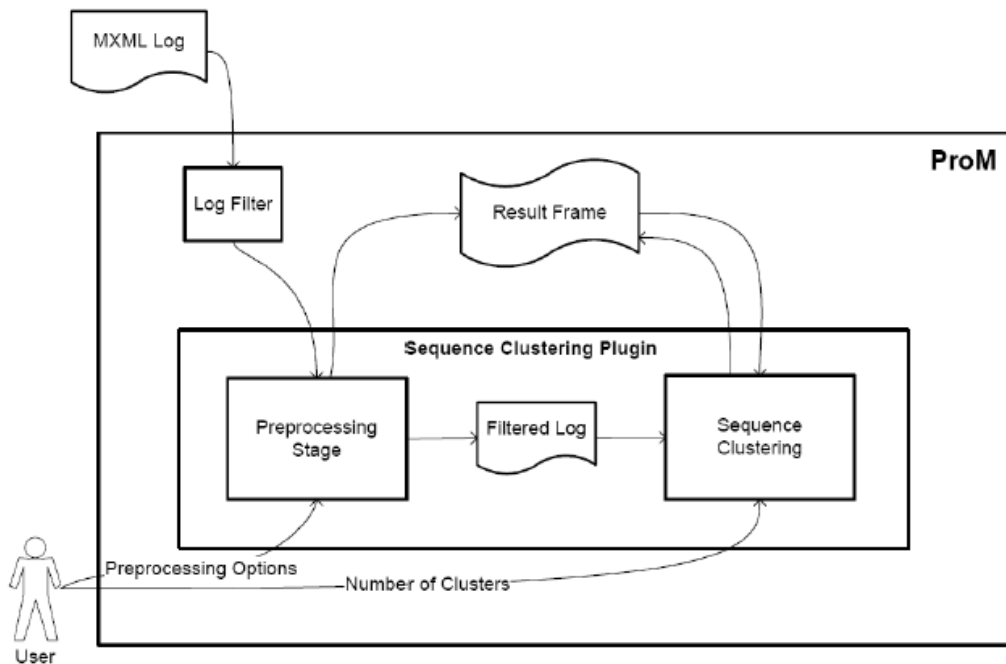


Figura 5. *Plug-in* de clusterização de seqüências do framework ProM [Veiga and Ferreira 2009].

Levando em consideração que processos no mundo real geralmente envolvem uma significativa quantidade de comportamentos não estruturados e ad-hoc (*Spaguetti models*), técnicas de pré-processamento baseado no algoritmo de clusterização de seqüências tem sido utilizada para resolver este problema, pois é uma técnica muito útil para descobrir padrões de comportamento e visualizá-los. Não obstante, os resultados do *clustering* de seqüências ainda vem sendo pesquisados para a comparação com outros métodos de clusterização.

4. Abordagem de Integração Proposta

Para melhorar o ciclo evolutivo dos processos de uma organização, A partir dos eventos gerados através de simulação baseada em agentes serão desenvolvidas técnicas apropriadas de mineração que sejam capazes de analisar e construir novos modelos de processos.

Considerando que a abordagem do *framework* de simulação AOR gera um *log* de eventos em formato *Extensible Markup Language* (XML), conforme apresentado na Figura 6, o mesmo pode servir como fonte de eventos como base para a mineração de processos.

A proposta então considera realizar uma transformação *eXtensible Stylesheet Language for Transformation* (XSLT) no *Log* de eventos gerados pelo simulador AOR para


```

<SimulationStep stepTime="1" xmlns:ns2="http://aor-simulation.org" xmlns="http://aor-simulation.org/log">
  <EnvironmentSimulatorStep>
    <ExogenousEvent nextOccurrenceTime="51.0" type="CustomerArrival">
      <ResultingStateChanges resultingFromRule="CustomerArrival_Rule">
        <Objects>
          <Obj id="6">
            <Slot value="1" property="lengthOfQueue"/>
          </Obj>
        </Objects>
        <Create>
          <Objects>
            <Obj id="-1" type="Customer" name="customer">
              <Slot value="false" property="waitingForPWFfree"/>
              <Slot value="1" property="arrivalTime"/>
              <Slot value="Null" property="order"/>
            </Obj>
          </Objects>
        </Create>
        <Collections>
          <Coll newSize="1" action="add" id="3" name="MenuBoardQueue">
            <Obj id="-1" type="Customer" name="customer"/>
          </Coll>
        </Collections>
      </ResultingStateChanges>
      <ResultingEvents resultingFromRule="CustomerArrival_Rule">
        <CausedEvent delay="1" type="StartMenuBoardService"/>
      </ResultingEvents>
    </ExogenousEvent>
  </EnvironmentSimulatorStep>
</SimulationStep>
<SimulationStep stepTime="2" xmlns:ns2="http://aor-simulation.org" xmlns="http://aor-simulation.org/log">

```

Figura 6. Exemplo do *Log* de eventos gerado pelo *framework* AOR.

possibilitar o pré-processamento do *log* transformando-o em estruturas de dados adequadas para aplicação de técnicas de mineração de processos, para descoberta e redefinição de processos. A Figura 7 ilustra a estratégia a ser utilizada para evolução dos processos.

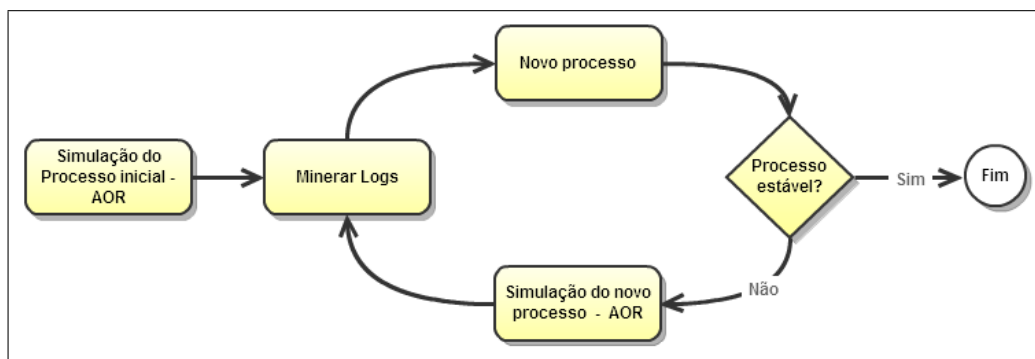


Figura 7. Estratégia para evolução dos processos.

Além das técnicas apresentadas na Seção 3, outras técnicas de mineração podem ser formuladas como hipóteses de trabalho no sentido de resolver os problemas anteriormente citados. Neste contexto, sugere-se também utilizar CEP – que é uma técnica que objetiva analisar uma série de dados em tempo real e com base nisso, identificar padrões e gerar eventos que podem ser tratados.

Tais abordagens podem ser utilizadas para mineração tanto em *log* de eventos, como em registros em banco de dados, banco de conhecimento, documentação de siste-

mas, especificação de requisitos entre outras fontes de informações capazes de proporcionar a registros de eventos de processos.

A simulação baseada em agentes pode proporcionar a geração das informações sem que empresas necessitem coletar as informações que serão a base para a mineração em ambientes de produção, podendo assim simular execuções de processos gerados através da integração entre os conceitos mencionados.

5. Conclusão

Este trabalho apresentou uma revisão de abordagens de modelos de processos de negócio e de técnicas de simulação baseada em agentes e mineração de processos.

Como técnicas que provêm suporte para projetar processos de negócio não fornecem uma semântica formal de comportamento, a utilização de técnicas de simulação baseada em agentes, através do uso do simulador AOR, pode ser uma abordagem interessante para proporcionar um processo de melhoria no modelo de aplicação, ainda mais se integrado a técnicas de mineração de processos.

A mineração de processos através de diversas abordagens, tais como a abordagem probabilística com base nos modelos matemáticos da cadeia de Markov de primeira ordem, *α -algorithm*, *heuristics miner*, *genetic miner*, *fuzzy miner* e clusterização de seqüências, demonstrou que é possível redescobrir o modelo de processo a partir da execução de instâncias de processo registradas em *log* de eventos.

Para as hipóteses de trabalho dos problemas apresentados, foi proposta a utilização do AOR como fonte de eventos para a mineração, i.e, cria-se o modelo de processo no AOR, realiza-se a simulação e depois são desenvolvidas técnicas que sejam capazes de analisar esses eventos e redefinir o modelo de processo. Como neste caso já existe um modelo de partida, e o AOR possibilita a geração dos eventos, o objetivo é desenvolver técnicas de mineração apropriadas e utilizar o modelo conhecido de partida para validação das técnicas utilizadas.

Entendemos que uma abordagem integrada de simulação baseada em agentes com mineração de processos possibilite verificar a existência de convergência no processo de definição e redescoberta de processos de negócio nas organizações. Os processos resultantes poderão auxiliar na busca dos objetivos principais da organização de forma mais eficiente, proporcionando respostas mais rápidas aos clientes, promovendo a utilização mais eficaz dos recursos e a conseqüente melhoria na qualidade do trabalho.

Como trabalho futuros, utilizando a combinação de implementação destas abordagens, pode-se propor estratégias de melhorias em processos de software que utilizam modelos e padrões de qualidade de software, (e.g, *Capability Maturity Model Integration - CMMI* [CMMI Product Team, Software Engineering Institute 2006], *Internacional Organization for Standardization - ISO Standards*, entre outros) como guia na definição de seus processos.

Ainda como trabalho futuro podemos citar o estudo mais aprofundado da proposta de integração de simulação baseada em agentes com diferentes técnicas de mineração de processos, realizando um estudo de caso real de um modelo de processo organizacional para auxílio na redescoberta dos processos.

Referências

- Barros, A., Decker, G., and Grosskopf, A. (2007). Complex events in business processes. *Business Information Systems*, pages 29–40.
- CMMI Product Team, Software Engineering Institute (2006). *CMMI for Development v1.2*. Carnegie Mellon University.
- Davenport, T. H. (1993). *Process Innovation – Reengineering Work through Information Technology*. Harvard Business School Press.
- Diaconescu, I. M. and Wagner, G. (2009). Agent-based simulations with beliefs and sparql-based ask-reply communication. In di Tosto, G. and Parunak, H. V. D., editors, *MABS*, volume 5683 of *Lecture Notes in Computer Science*, pages 86–97. Springer.
- Eckert, M. and Bry, F. (2009). Complex event processing (cep). *Informatik-Spektrum*, 32(2):163–167.
- Eriksson, H. E. and Penker, M. (2000). *Business Modeling With UML: Business Patterns at Work*. Wiley, 1 edition.
- Ferreira, D. R. and Gillblad, D. (2009). Discovering process models from unlabelled event logs. In Dayal, U., Eder, J., Koehler, J., and Reijers, H. A., editors, *Business Process Management, 7th International Conference, BPM 2009, Ulm, Germany, September 2009. Proceedings*, volume 5701 of *Lecture Notes in Computer Science*, pages 143–158. Springer.
- Ferreira, D. R., Zacarias, M., Malheiros, M., and Ferreira, P. (2007). Approaching process mining with sequence clustering: Experiments and findings. In Alonso, G., Dadam, P., and Rosemann, M., editors, *BPM*, volume 4714 of *Lecture Notes in Computer Science*, pages 360–374. Springer.
- Georgakopoulos, D., Hornick, M. F., and Sheth, A. P. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2):119–153.
- Günther, C. W. and van der Aalst, W. M. P. (2007). Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In Alonso, G., Dadam, P., and Rosemann, M., editors, *BPM*, volume 4714 of *Lecture Notes in Computer Science*, pages 328–343. Springer.
- Hammer, M. and Champy, J. (2003). *Reengineering the corporation*. HarperBusiness Essential. HarperBusiness, New York, NY, rev. and updated with a new authors' note edition.
- Laguna, M. and Marklund, J. (2004). *Business process modeling, simulation and design*. Pearson-Prentice hall, Upper Saddle River, New Jersey.
- Medeiros, A. A., Weijters, A., and van der Aalst, W. (2007). Genetic process mining: An experimental evaluation. *Journal of Data Mining and Knowledge Discovery*, 14(2):245–304.
- OMG (2008). *Business Process Model and Notation*. Object Management Group, v1.1 edition.

- Pascalau, E., Giurca, A., and Wagner, G. (2009). Validating auction business processes using agent-based simulations. In Abramowicz, W., Maciaszek, L. A., Kowalczyk, R., and Speck, A., editors, *BPSC*, volume 147 of *LNI*, pages 95–109. GI.
- Song, M., Günther, C. W., and van der Aalst, W. M. P. (2008). Trace clustering in process mining. In Ardagna, D., Mecella, M., and Yang, J., editors, *Business Process Management Workshops*, volume 17 of *Lecture Notes in Business Information Processing*, pages 109–120. Springer.
- van der Aalst, W., Hofstede, A. T., and Weske, M. (2003). Business process management: A survey. page 1019.
- van der Aalst, W. M. P., Weijters, T., and Maruster, L. (2004). Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1128–1142.
- van Dongen, B. F., Medeiros, A. K. A., Verbeek, H. M. W., Weijters, A. J. M. M., and van der Aalst, W. M. P. (2005). The prom framework: A new era in process mining tool support. In Ciardo, G. and Darondeau, P., editors, *ICATPN*, volume 3536 of *Lecture Notes in Computer Science*, pages 444–454. Springer.
- Veiga, G. M. and Ferreira, D. R. (2009). Understanding spaghetti models with sequence clustering for prom. In Rinderle-Ma, S., Sadiq, S. W., and Leymann, F., editors, *Business Process Management Workshops*, volume 43 of *Lecture Notes in Business Information Processing*, pages 92–103. Springer.
- Wagner, G. (2003). Aor modelling and simulation: Towards a general architecture for agent-based discrete event simulation. In Giorgini, P., Henderson-Sellers, B., and Winiakoff, M., editors, *AOIS*, volume 3030 of *Lecture Notes in Computer Science*, pages 174–188. Springer.
- Wagner, G. (2009). business rules and agent-based business process simulation. Simpósio Brasileiro de Sistemas de Informações (SBSI) - Brasília-DF.
- Wagner, G. and Diaconescu, I. M. (2009). Aor-simulation.org: cognitive agent simulation. In Sierra, C., Castelfranchi, C., Decker, K. S., and Sichman, J. S., editors, *AAMAS* (2), pages 1405–1406. IFAAMAS.
- Walicki, M. and Ferreira, D. R. (2010). Mining sequences for patterns with non-repeating symbols. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE.
- Weijters, A., van der Aalst, W., and de Medeiros, A. A. (2006). Process mining with the heuristicsminer algorithm. *BETA Working Paper Series WP 166*, Eindhoven University of Technology.
- Wooldridge, M. (2009). *An Introduction to Multiagent Systems*. Wiley, Chichester, UK, 2. edition.