

Uma interpretação para os elementos de EPCs com base em uma ontologia de fundamentação

Paulo Sérgio Santos Jr., João Paulo A. Almeida, Giancarlo Guizzardi

Departamento de Informática – Universidade Federal Espírito Santo (UFES)

Av. Fernando Ferrari, s/n, Vitória, ES, Brasil

paulossjunior@inf.ufes.br; jpalmeida@ieee.org; gguizzardi@acm.org

RESUMO

Este trabalho apresenta uma análise ontológica da linguagem de modelagem de processos EPC (Event-driven Process Chains) como implementada no ARIS Toolset. A abordagem de análise ontológica permite realizar a interpretação das entidades de EPCs em termos de uma ontologia de fundamentação, permitindo também, identificar problemas que afetam a clareza, expressividade e outras características da linguagem de modelagem. Como ponto de partida, utilizamos um metamodelo da linguagem de modelagem de processos do ARIS Method previamente identificado em trabalhos anteriores.

ABSTRACT

This paper presents an ontological analysis of the EPC (Event-driven Process Chain) metamodel as implemented in the ARIS Toolset. The ontological analysis allows us to interpret modeling elements in EPC diagrams in terms of a foundational ontology. This enables us to define the precise real-world semantics for EPCs and allows us to identify problems affecting the clarity and expressiveness of EPCs. As a starting for our analysis, we adopt a metamodel of the ARIS Method that we have defined in our earlier work.

Keywords

Análise ontológica, Ontologias de Fundamentação, Processo de negócio, EPC, ARIS, ARIS Method.

1. INTRODUÇÃO

O ARIS (*ARchitecture for integrated Information Systems*) é um framework de modelagem organizacional que tem como objetivo principal permitir a descrição e o desenvolvimento de sistemas de informação integrados à estrutura de uma organização por meio de seus processos de negócio [2][3].

Esse framework é estruturado em diferentes visões (organizacional, dados, controle, função e saída) e camadas de abstração [2][3]. Neste trabalho, nos concentramos na visão de controle dos níveis mais altos de abstração. A *visão de controle* descreve os processos de transformação da informação por meio de uma função ou um conjunto de funções. Como as funções representam atividades organizacionais potencialmente complexas, esta visão é usada para modelar processos de negócio na abordagem ARIS.

Cada uma das visões da abordagem ARIS possui uma linguagem própria, que pode ser definida através de sua sintaxe e semântica [2][3]. A linguagem utilizada na *visão de controle* do ARIS

Method para a modelagem de processos de negócio é o EPC (*Event-Driven Process Chains*) [2][3].

A importância de EPCs na modelagem de processos pode ser comprovada pela existência de diversas ferramentas comerciais que utilizam EPCs, por exemplo, WebSphere [8], ARIS Toolset entre outros da IDS Scheer [11]. Devido ao valor econômico desta linguagem, diversos trabalhos acadêmicos têm sido realizados, como, por exemplo, trabalhos na área de análise ontológica [9][21], na área de transformações entre linguagens utilizando técnicas de Desenvolvimento Orientado a Modelos [11][14][15][16] e outros.

Diversos trabalhos na literatura propõem semânticas formais para EPCs utilizando Redes de Petri [24][18][23][12], com o objetivo de demonstrar que não há problemas estruturais na linguagem e, desta forma, assegurar que processos de negócio especificados em EPCs são passíveis de automatização.

Outros trabalhos visam definir a semântica e a correte de EPCs através de interpretações ontológica como, por exemplo, em [9] e [21] que possuem como objetivo, respectivamente, definir a semântica do EPC utilizando ontologias de fundamentação e ontologias de processo de negócio.

Apesar dos trabalhos supracitados terem definido a semântica formal e ontológica para o EPC, tais trabalhos não foram realizados sobre a linguagem implementada na ferramenta de modelagem ARIS Toolset. Como demonstramos em [19], o metamodelo da linguagem nesta ferramenta é significativamente diferente da proposta original de Scheer [2]. Desta forma, os trabalhos citados anteriormente não demonstram a correte, clareza e outros aspectos da linguagem de modelagem processos de fato implementada.

Para suprir tal deficiência este artigo tem por objetivo apresentar uma análise ontológica que viabiliza: (i) a definição formal e rigorosa do EPC implementado nas ferramentas de modelagem em termos de entidades do mundo real; (ii) a identificação de elementos inadequados da linguagem de modelagem baseada em EPCs utilizando o framework de avaliação de linguagens apresentado em [6]; e (iii) indicar recomendações de melhoria na linguagem caso sejam identificados problemas na interpretação ontológica dos elementos da linguagem.

O artigo está organizado da seguinte forma: a seção 2 apresenta o metamodelo do EPC que será analisado, a seção 3 apresenta os conceitos ontológicos utilizados na análise, a seção 4 fornece uma interpretação para cada elemento do metamodelo. Finalmente, a seção 5 apresenta as conclusões e trabalhos futuros.

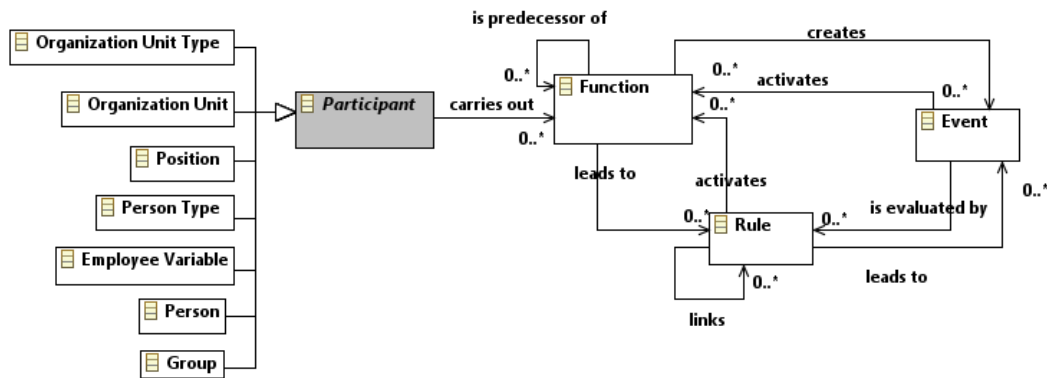


Figura 1 - Fragmento do metamodelo da linguagem de processo do ARIS Method, baseado em [19].

2. METAMODELO DA LINGUAGEM DE PROCESSOS DE NEGÓCIO DO ARIS METHOD (EPC)

A Figura 1 apresenta um fragmento do metamodelo da linguagem de processo de negócio do ARIS Method (como implementado no ARIS Toolset). O metamodelo foi construído a partir da abordagem proposta em [19]. As principais metaclasses deste fragmento são: *Participant*, *Event*, *Rule* e *Function*. A Figura 2 apresenta um exemplo de modelo de processo de negócio construído através da linguagem EPC, com elementos de modelagem que instanciam o metamodelo apresentado na Figura 1.

A metaclasses *Participant* é uma metaclasses abstrata que subsume os elementos organizacionais *Organization Unit Type*, *Organization Unit*, *Position*, *Person Type*, *Person*, *Group* e *Employee variable*. A semântica de cada um destes elementos é apresentada em [19]. Aqui, basta considerar que a metaclasses *Participant* captura a relação comum que os diversos elementos organizacionais têm como o elemento *Function*. Na Figura 2, os elementos “Cliente” e “Vendedor” são instâncias da metaclasses *Position* e *Person Type*, respectivamente.

A metaclasses *Function* é elementar para a modelagem de processos com EPCs. De acordo com a documentação on-line do ARIS Toolset, o elemento *Function* representa uma atividade técnica ou atividade realizada sobre um determinado objeto, que possui como finalidade apoiar um ou vários objetivos de negócios da organização [2]. Uma atividade pode ser realizada por uma pessoa ou por um sistema computacional [2]. A atividade possui entradas (do inglês, *inputs*) como, por exemplo, informações ou matéria-prima e criam saídas (do inglês, *outputs*) com, por exemplo, informações diferentes ou produtos. As atividades também podem consumir e criar recursos organizacionais em sua execução [3]. Na Figura 2, “Solicitar compra”, “Analisar solicitação de compra”, “Finalizar compra” e “Informar Cliente”

são exemplos de atividades (instâncias da metaclasses *Function*)

O relacionamento *carries out* entre o elemento *Participant* e *Function* representa que um ou mais participantes do processo de negócio são os responsáveis pela execução da atividade. Na Figura 2, este relacionamento é representado através da posição das atividades nas raias correspondentes a uma instância de *Participant*.

O auto-relacionamento *is predecessor of* do elemento *Function* representa que uma atividade que predecessor de outra atividade do processo de negócio. Assim, é possível expressar que a atividade “Solicitar compra” acontece antes da atividade “Analisar solicitação de compra”, como pode ser observado na Figura 2.

Um evento (*Event*) representa um estado que é relevante para o gerenciamento do processo e influencia o fluxo de execução de um processo de negócio. Mudanças no estado são refletidas na troca do status da informação ou das informações relevantes do processo. Eventos ativam atividades (*Function*) e são resultados de atividades ou são criados por atores externos ao processo [2][3]. Segundo [3], os eventos, representam as pré-condições e pós-condições para cada etapa do processo. As pré-condições representam um estado da realidade que ativam uma ou mais atividades, enquanto as pós-condições representam um estado da realidade após a execução da atividade. A Figura 2 apresenta vários exemplos de instâncias da metaclasses *Event*, representados como hexágonos (por exemplo, “Necessidade de compra identificada”).

Os relacionamentos *activates* e *creates* entre *Function* e *Event* representam, respectivamente, que uma atividade é ativada por um ou mais eventos e que uma ou mais atividades podem criar um ou mais eventos. As instâncias destes relacionamentos são representadas através de setas entre atividades e eventos na Figura 2.

O elemento *Rule* é utilizado para controlar o fluxo do modelo de

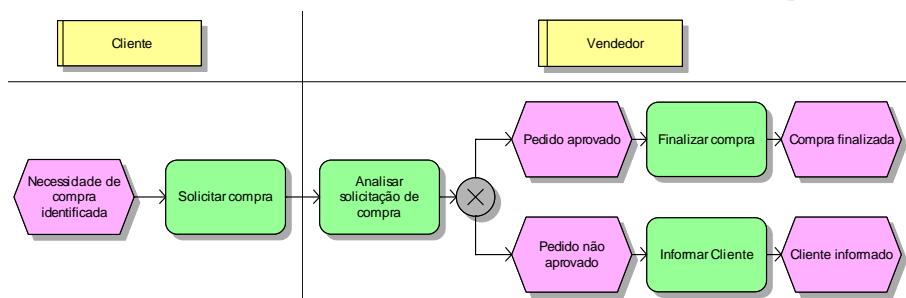


Figura 2 – Exemplo de Processo de Negócio modelado em EPC.

processo. Uma Rule define como o processo deve seguir, tomando como base o resultado e o efeito de atividades que a precedem [3]. Neste contexto, o relacionamento is evaluated by é utilizado para indicar quais eventos são considerados em uma Rule. O relacionamento activates entre Function e Rule especifica quais Functions são atividades pela Rule. O relacionamento leads to representa a relação entre a atividade que antecede a regra (Rule) e os eventos (Event) que são criados pela atividade. O auto-relacionamento links do elemento Rule é utilizado para especificar regras comportamentais do processo de negócio mais complexas. Todos são representados como setas em EPCs.

O ARIS Method possui três tipos básicos de Rule: XOR, AND e OR [3]. A Tabela 1 apresenta a definição destes elementos. Estas regras são utilizadas para criar joins e splits no processo de negócio. Um join é utilizado para unir diversos ramos do processo de negócio em apenas um ramo. O split é utilizado para dividir um ramo do processo de negócio em dois ou mais ramos. A Figura 2 apresenta um exemplo de XOR split, indicando que a atividade “Analisar compra” será seguida pela atividade “Finalizar compra” ou pela atividade “Informar Cliente”.

Tabela 1 – Descrição das Regras Básicas.

Operador	Símbolo	Formando um Join	Formando um Split
OR		Qualquer evento, ou combinação de eventos, irão ativar uma atividade	Um ou mais caminhos serão habilitados com o resultado da atividade
XOR		Um, e somente um, evento ativa uma atividade	Um, e somente um, caminho será habilitado com o resultado da atividade
AND		Somente após a execução de todos os eventos que a atividade será ativada	Divide o processo em dois ou mais caminhos em paralelo.

3. BASE ONTOLÓGICA

Para realizar a análise ontológica serão utilizados os conceitos das ontologias de fundamentação UFO-A, UFO-B e UFO-C [4] [5]. A Figura 3 apresenta um fragmento com os conceitos da UFO-A,

UFO-B e UFO-C inter-relacionados.

A UFO-B [5] apresenta o conceito de Events. Os Eventos acontecem no tempo, no sentido de que eles se estendem no tempo acumulando partes temporais. Um exemplo de um Evento é um processo de negócio. Sempre que um Event está presente, não é o caso que todas as suas partes temporais estarão presentes.

Eventos (Events) podem ser especializados em dois tipos: (i) atômicos (Atomic Event) ou (ii) complexos (Complex Event). O primeiro não possui subpartes, enquanto o segundo é composto de agregações de pelo menos dois outros eventos – que por sua vez podem ser atômicos ou complexos. Eventos produzem possíveis transformações de uma porção da realidade para outra, ou seja, eles podem mudar a realidade mudando o estado de uma (pre-state) Situation para uma (post-state) Situation.

Eventos são entidades ontologicamente dependentes uma vez que eles dependem existencialmente de outros participantes (Participants) para que existam. Um exemplo de Event é a “Conversa entre médico e paciente”. Neste evento há a participação de dois indivíduos: o médico e o paciente. Cada uma destas participações é por si só um evento, que pode ser atômico ou complexo, mas existencialmente dependente de um Substantial.

As Situations (situações) são tipos especiais de endurants [4][5][6]. Situations são entidades complexas que são constituídas possivelmente por muitos endurants (incluindo outras Situations). Situations são assumidas aqui como sinônimos do que é denominado na literatura de state of affairs, isto é, uma parcela da realidade que possa ser compreendida como um todo. Os exemplos de situações incluem “Paulo possui um carro” e “Paula está acessando o computador de Paulo enquanto ele está assistindo televisão”. Um Situation Universal [4] é usado para definir situações com propriedades em comum.

A UFO-C [5] adiciona distinções relativas à intencionalidade de eventos de UFO-B. UFO-C introduz os conceitos de Action, Action Universal, Action Contribution, Agent e outros. Uma Action é um evento intencional, ou seja, um evento que instancia um plano (Action Universal) com o propósito específico de satisfazer algum compromisso interno (Internal Commitment) de um agente. Somente agentes – entidades capazes de ter intentional moments[5] – podem executar ações. Uma Action pode ser composta de outras Actions. Uma

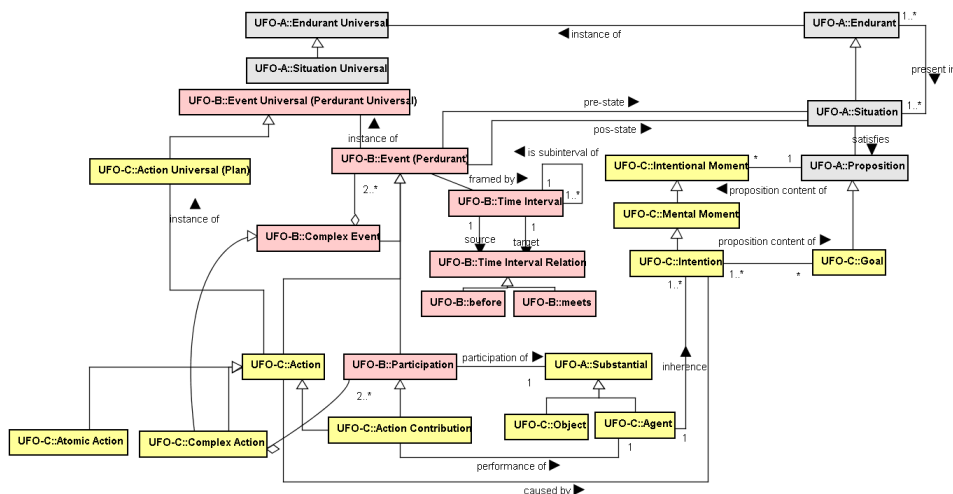


Figura 3 – Fragmento da UFO-A, UFO-B e UFO-C com os conceitos inter-relacionados.

Action composta de duas ou mais Actions é chamada de ação complexa (Complex Action), enquanto uma Action que não possui partes é chamada de ação atômica (Atomic Action). Uma ação complexa pode ser composta tanto por outras ações complexas quanto por ações atômicas.

Uma ação complexa é composta de duas ou mais participações (Participation). As participações podem ser eventos intencionais ou eventos não-intencionais. Um exemplo de uma ação complexa é “o ato de escrever este artigo no computador”. Esta ação é composta pelas participações intencionais dos autores e uma participação não-intencional do computador. Quando um agente (entidades capazes de ter intentional moments) participa de uma ação, esta ação é chamada de Action Contribution. Uma ação complexa que é composta de action contributions de diferentes agentes, é chamada de Interaction (Interação).

Na UFO-B, define-se o espaço conceitual de tempo como sendo uma estrutura “composta de” time intervals (intervalos de tempo). Um intervalo de tempo, por sua vez, é “composto de” Time points (pontos no tempo). Para representar especificamente uma estrutura de tempo linear e ordenada, tais pontos no tempo podem ser representados como números reais e os intervalos de tempo como sendo um conjunto de números reais. Neste modelo de tempo podem ser utilizadas as chamadas Allen’s Relations [1]. As diversas relações possíveis entre eventos a e b são representadas esquematicamente na Figura 4, na qual a direção horizontal representa a passagem do tempo.

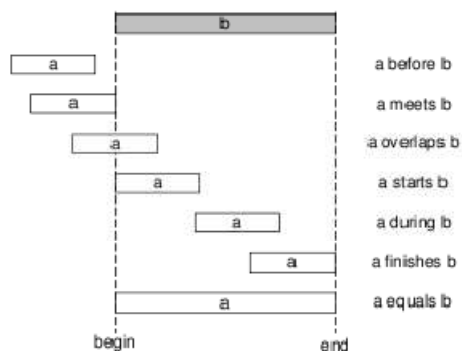


Figura 4 - Relações temporais entre um par de eventos [1].

4. ANÁLISE ONTOLÓGICA DO METAMODELO DE PROCESSO DE NEGÓCIO DO ARIS METHOD

De acordo com [20] e [17], um processo de negócio pode ser definido como uma coleção de atividades organizacionais inter-relacionadas, iniciadas em resposta a um evento, que visam alcançar algum objetivo organizacional. Em outras palavras, um processo de negócio é uma especificação de uma funcionalidade organizacional que deve ser realizada para alcançar um determinado objetivo. Um processo de negócio pode ser modelado como um EPC. Desta forma, podemos dizer que, coletivamente, os elementos de um EPC podem ser interpretados como um Action Universal em UFO-C.

4.1 Análise ontológica do elemento Function

De acordo com [2], uma Function pode ser definida em vários níveis de abstração e refinamento. Assim, é possível interpretar o elemento Function de diferentes formas: (i) Atomic Action

Universal, (ii) Complex Action Universal ou (iii) Interaction Universal.

A interpretação do elemento Function como um Atomic Action Universal ocorre quando o elemento Function representa uma ação atômica (não possui outros níveis de refinamento) e é executada por somente um agente (Participant).

A interpretação do elemento Function como um Complex Action Universal ocorre quando o elemento Function representa uma ação não-atômica, composta de dois ou mais Action Universals, e é executada por somente um agente (Participant).

A interpretação do elemento Function como Interaction Universal ocorre quando o elemento Function representa uma Action (Atomic Action ou Complex Action) que é executada por dois ou mais agentes.

Dévido às diferentes interpretações ontológicas do elemento Function é possível identificar um problema de *incompleteness* na linguagem, i.e., existem entidades na ontologia de referência que não são representadas na linguagem. Desta forma, há uma sobrecarga semântica no elemento Function, gerando, assim, o problema de não-lucidez na linguagem de modelagem. Uma possível solução deste problema é a criação de novos elementos notacionais que representem as diferenças ontológicas do elemento Function.

A existência de um relacionamento carries out entre um Participant p e uma Function f indica que um agente participará intencionalmente da atividade organizacional representada por f . Este agente contribuirá para a atividade organizacional representada por f intencionalmente através de uma Action Contribution. Dependendo da metaclasses que especializa p , o agente será uma instância do universal representado por p (nos casos de Position, Person Type, Organizational Unit Type) ou será o próprio agente (Person, Organization Unit).¹

O auto-relacionamento is predecessor of do elemento Function representa um relacionamento temporal entre duas ou mais atividades do processo de negócio, ou seja, este relacionamento define que para que uma ou mais atividades sejam executadas é necessário que a atividade, antecessora a essas, tenha sido concluída. Esse relacionamento também descreve que uma atividade “produz” um pré-estado (Situation) de uma atividade sucessora a essa.

Por fim, o relacionamento is predecessor of é interpretado como uma Time Relation da UFO-B, mais especificamente, uma “before” relation ou uma “meets” relation.

4.2 Interpretação do elemento Event

O elemento Event é utilizado de duas formas distintas em EPCs: para representar uma situação de interesse (informalmente, um “estado” de interesse) ou para representar uma ação arbitrária que influencia a execução do processo (informalmente um “evento”).

No primeiro caso, o elemento Event do ARIS Method deve ser interpretado como um Situation Universal. Durante a execução do processo, instâncias deste Universal serão os pré-estados (situações) que habilitam atividades organizacionais ou pós-estados (situações) que resultam destas. Como eventos no

¹ A interpretação de um participante representado pela metaclasses Group envolve o uso de outros conceitos de UFO que não são abordados neste artigo por limitações de espaço. Já a documentação do elemento Employee variable não dá subsídios para uma interpretação ontológica.

ARIS podem envolver uma descrição textual arbitrária, não há regras sobre que tipos de entidades podem estar presentes (present in) na situação. Um evento inicial é denotado pela ausência de atividades do processo que a antecedam (ou seja, não há Function relacionada através de creates com um evento inicial). Os eventos iniciais denotam, portanto as situações que habilitam o processo como um todo. Atividades que habilitem esta situação não fazem parte do processo. Similarmente, um evento final é denotado pela ausência de atividades do processo que o sucedam.

No segundo caso, o elemento Event do ARIS Method é utilizado para representar ocorrências eventos (intencionais ou não intencionais) que dependem de um Substantials. Por exemplo, “o clicar de mouse” ou “mudança de temperatura”. Desta forma, o elemento Event do ARIS Method é interpretado como um Event Universal da UFO. Como eventos no ARIS podem envolver uma descrição textual arbitrária, não há regras sobre que tipos de entidades podem participar do Event (ou seja, que tipos de Participations são permitidas).

Para o uso do elemento Event do ARIS Method como um Situation Universal, o relacionamento activates representa que um ou mais eventos são pré-estados (pre-state Situation) necessários para uma ou mais atividades. Em outras palavras, para que a atividade inicie sua execução é necessário que a situação exista. O relacionamento creates entre Function e Event representa que um ou mais atividades geram situações (post-state da Function) que podem ser pré-estados para outras atividades do processo de negócio.

Caso seja considerada a interpretação que o elemento Event do ARIS Method é um Event Universal, o relacionamento activates representa um relacionamento temporal, mais especificamente um “Before” ou “Meets”, entre os Events representados por Event e Function. Ou seja, este relacionamento define que para que uma ou mais atividades sejam executadas é necessário que os eventos antecessores a essas atividades tenham ocorrido. O relacionamento creates representa o inverso de activates (também uma relação entre Events).

Devido às diferentes interpretações ontológicas, o elemento Event do ARIS Method apresenta o problema de sobrecarga semântica. Uma possível solução deste problema é a criação de novos elementos notacionais que representem as diferenças ontológicas do elemento Event. Por exemplo, criar elementos na linguagem para representar eventos intencionais e não intencionais. Outra recomendação possível é o uso exclusivo de Events para representar “estados” (como sugerido por [3]). Desta forma, não há sobrecarga semântica.

4.3 Análise ontológica do elemento Rule

O conjunto de Rules presente em um processo de negócio especifica quais atividades devem ser executadas, tomando como base o estado das coisas (do inglês, *state of affairs*). O estado das coisas para o processo de negócio é construído a partir dos eventos internos (produzidos pelas atividades) e dos eventos externos (produzidos por outros processos ou por outras entidades) ao processo de negócio.

Através do elemento Rule é possível compor eventos complexos a partir da união de eventos internos do processo de negócio e/ou eventos externos ao processo de negócio. Estes eventos complexos podem ser pré-requisitos para que uma determinada atividade do processo ocorra. Por exemplo, o elemento notacional

AND é utilizado para compor um evento complexo, a partir de dois ou mais eventos do processo de negócio, que pode ser o pré-estado de uma ou mais atividade do processo de negócio. O auto-relacionamento Links do elemento Rule também pode ser utilizado para criar eventos complexos através de combinações de Rules.

Diferentemente dos demais elementos do metamodelo de processo de negócio do ARIS Method, o elemento Rule não apresenta uma interpretação direta para uma entidade ontológica da UFO. Isto ocorre, porque os Rule descrevem relações entre universais que restringem possíveis relações entre instâncias. Por exemplo, assumindo que o elemento Event represente uma Situation, temos as seguintes interpretações: (i) X, Y AND Z: significa que existe um Situation Universal W tal que toda instância de Z tem como pré-situação uma instância w de W e toda instância de w é a soma mereológica de uma instância de X e uma instância de Y (ambas situações); (ii) X, Y OR Z: significa que toda instância de Z tem como pré-situação uma instância w tal que w é uma instância de X ou de Y ou a soma mereológica ambas instâncias; e (iii) X, Y XOR Z é igual a (ii) com o requisito adicional de que X e Y sejam Situation Universals que são disjuntos.

5. CONCLUSÃO E TRABALHOS FUTUROS

A abordagem de análise ontológica apresentada neste artigo possibilitou um melhor entendimento dos conceitos presentes em EPCs com base em uma ontologia de fundamentação.

Um benefício direto da análise ontológica do EPC está relacionado ao desenvolvimento de modelos de processo com semântica bem definida. Como os EPCs são usados para capturar e comunicar conhecimento sobre uma organização, além de serem usados para a construção de sistemas de informação, é necessário que não haja ambiguidades em seus elementos. Com uma semântica bem definida, é possível utilizar EPCs como base para o desenvolvimento de sistemas orientados a processos.

A avaliação ontológica permitiu também revelar problemas de uso (como o caso do uso do elemento Event). A análise permitiu identificar de forma sistemática, um problema já identificado informalmente na literatura ([3], p. 111).

Um dos trabalhos que mais se assemelham a este é o apresentado por Green e Rosemann em [6]. Green e Rosemann discutem a avaliação ontológica do EPC com base na ontologia BWW [25]. O nosso trabalho difere do trabalho de Green e Rosemann em três aspectos principais.

Primeiramente, utilizamos como ponto de partida para a nossa avaliação um fragmento do metamodelo que representa fielmente a linguagem de modelagem de processos em EPCs como implementado na ferramenta ARIS Toolset. Desta forma, a avaliação se aplica aos modelos produzidos nesta ferramenta. Em sua vez, Green e Rosemann consideraram o metamodelo apresentado em Scheer[2] que já não retrata a linguagem como adotada nas organizações que utilizam a ferramenta ARIS Toolset [19].

Adicionalmente, Green e Rosemann interpretam apenas os elementos da linguagem isoladamente. Neste trabalho, avaliamos também a interpretação dos relacionamentos entre os elementos da linguagem (meta-associações no metamodelo).

Finalmente, utilizamos uma ontologia de fundamentação diferente na análise ontológica. Isto nos permitiu chegar a diferentes

conclusões com relação à interpretação de alguns elementos de modelagem. Por exemplo, nossa interpretação de Function difere significativamente daquela apresentada em [6]. Naquele trabalho, uma Function é interpretada como um mapeamento formal entre dois valores em um espaço conceitual de estados. Desta forma Green e Rosemann não fornecem uma explicação ontológica, por exemplo, para : (i) participação de agentes em atividades organizacionais; (ii) noção de tempo associada à execução de uma atividade organizacional; (iii) definição de funções complexas através de relações mereológicas entre funções e sub-funções; e (iv) dentre outras relações que envolvem funções, eventos, agentes e compromisso.

Os próximos passos deste trabalho se concentrarão na análise ontológica das linguagens de domínio organizacional, de detalhamento de atividades (Function Allocation Diagram - FAD) e de outras linguagens providas pelo ARIS Method.

6. REFERÊNCIAS

- [1] ALLEN, J.F. 1983. Maintaining Knowledge About Temporal Intervals. *Communications of the ACM*, Vol.26, no. 11.
- [2] ARIS - Scheer, A.W. 1999. ARIS – Business Process Modeling. Third Edition. Springer.
- [3] Davis, R. 2001. Business Process Modeling with ARIS - A Practical Guide. Springer.
- [4] Dockhorn, P. 2006. Situations in Conceptual Modeling of Context. *Proceedings of the IEEE EDOC 2nd International*. Hong Kong. *Ontologies and Rules for The Enterprise (VORTE'06)*.
- [5] Guizzardi, G., Falbo, R. A., Guizzardi, R. S. S. 2008. The Importance of Foundational Ontologies for Domain Engineering: The case of the Software Process Domain (A importância de Ontologias de Fundamentação para a Engenharia de Ontologias de Domínio: o caso do domínio de Processos de Software – in portuguese). *IEEE Transactions Latin America*.
- [6] Guizzardi, G.; Ferreira Pires, L.; Van Sinderen, M. 2005. An Ontology-Based Approach for Evaluating the Domain Appropriateness and Comprehensibility Appropriateness of Modeling Languages. *ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems*, LNCS vol. 3713, Springer-Verlag.
- [7] Heller, B., Herre, H. 2004. **Ontological categories in gol. Axiomathes** – Kluwer Publishers, v. 14, p. 71–90.
- [8] IBM. “XML Import tool from ARIS to IBM WebSphere Business Modeler”. www.alphaworks.ibm.com/tech/arisimport/requirements.
- [9] Islay, D. G., Rosemann, M., Green, P. 2004. “Exploring proposed ontological issues of ARIS with different categories of modellers”. In: *Proceedings of the Australasian Conference on Information Systems*, Hobart, Australia.
- [10] Jonkers, H., Lankhorst, M., Van Buuren, R.S., Hoppenbrouwers, M.M. B, Van Der Torre, L. 2004. Concepts for modeling enterprise architectures. In special issue on Architecture in IT of the *International Journal of Cooperative Information Systems*, 13(3):257--287. World Scientific, Setembro.
- [11] Kern, H., Kühne, S. 2007. Model Interchange between ARIS and Eclipse EMF. In *The 7th OOPSLA Workshop Domain-Specific Modeling*, Computer Science and Information System Reports. Technical Reports, TR-38, University of Jyväskylä, Finland.
- [12] Kindler, E. 2006. On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. Technical Report. Computer Science Department, University of Paderborn, Germany.
- [13] Lankhorst, M. 2005. *Enterprise Architecture at Work - Modelling, Communication and Analysis*, Springer.
- [14] Mendling, J. and Nüttgen, M. 2004. Transformation of ARIS Markup Language to EPML. *Proc. of the 3rd GI Workshop on Event-Driven Process Chains*.
- [15] Mendling, J., Neumann, G., Nüttgens M. 2005. Yet Another Event-Driven Process Chain (Extended Version). Technical Report JM-2005-05-27. Vienna University of Economics and Business Administration
- [16] Mendling, J., Neumann, G., Nüttgens, M. 2005. Towards Workflow Pattern Support of Event-Driven Process Chains (EPC). In: M. Nüttgens, J. Mendling, eds.: *Proc. of the 2nd GI Workshop XML4BPM - XML for Business Process Management* at BTW 2005, Karlsruhe, Germany, CEUR Workshop Proceedings [Volume 145](http://www.ceur-ws.org), ISSN-1613-0073, pages 23-38, March.
- [17] Ouyang, C., Dumas, M., Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., Mendling, J. 2008. [From Business Process Models to Process-oriented Software Systems](http://www.sciencedirect.com/science/article/pii/S1569190X08700011): The BPMN to BPEL Way. *ACM Transactions on Software Engineering and Methodology*.
- [18] Rosemann, M., Van der Aalst, W.M.P. 2007. A configurable reference modelling language. *Inf. Syst.* 32, 1 (Mar. 2007), 1-23. DOI= <http://dx.doi.org/10.1016/j.is.2005.05.003>
- [19] Santos JR., P. S., Almeida, J. P. A. 2008. Escavando as Linguagens de Modelagem Organizacional e Modelagem de Processos de Negócio do ARIS Method. In: *Workshop em Gerência de Processos de Negócio (WBPM)*.
- [20] Sharp, A., McDermott, P. 2000. *Workflow modeling: Tools for Process Improvement and Application Development*. ISBN: 1-58053-021-4.
- [21] Thomas, O., Fellmann, M. 2006. *Semantic Event-Driven Process Chains*. In: *Workshop SBPM at the ESWC 2006*, Budva, Montenegro, Junho.
- [22] Van der Aalst, W.M.P. 1998. [The Application of Petri Nets to Workflow Management](http://www.sciencedirect.com/science/article/pii/S1569190X98700011). *The Journal of Circuits, Systems and Computers*, 8(1):21-66,
- [23] Van der Aalst, W.M.P. 1998. Formalization and Verification of Event-driven Process Chains. *Computing Science Reports* 98/01, Eindhoven University of Technology, Eindhoven.
- [24] Van Dongen, B.F., Van der Aalst, W.M.P., Verbeek., H.M.W. 2005. [Verification of EPCs: Using Reduction Rules and Petri Nets](http://www.sciencedirect.com/science/article/pii/S1569190X05700011). *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, volume 3520 of *Lecture Notes in Computer Science*. pages 372-386. Springer-Verlag, Berlin.
- [25] Weber, R. 1997. *Ontological Foundations of Information System*. Melbourne, Australia: Melbourne, Vic., Coopers & Lybrand and the Accounting Association of Australia and New Zealand.