

# Escavando as Linguagens de Modelagem Organizacional e Modelagem de Processos de Negócio do ARIS Method

Paulo Sérgio Santos Jr., João Paulo A. Almeida

Departamento de Informática – Universidade Federal Espírito Santo (UFES)

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

paulossjunior@gmail.com; jpalmeida@ieee.org

## RESUMO

Neste trabalho, propomos uma abordagem de escavação dos metamodelos das linguagens de modelagem organizacional e modelagem de processos de negócio presentes no ARIS Method. A abordagem utiliza informações obtidas a partir das interações do usuário com o ambiente de modelagem ARIS Toolset, assim como as informações contidas na documentação desta ferramenta. A aplicação da abordagem resulta em metamodelos das linguagens, esclarecendo os elementos de modelagem e os relacionamentos entre estes elementos. Os metamodelos servem como ponto de partida para a definição da semântica da linguagem e viabilizam ainda a construção de ferramentas de modelagem, simulação, análise e transformação de modelos organizacionais e de processos de negócio.

## ABSTRACT

In this paper we propose an excavation approach for the languages for organizational modeling and business process modeling in the ARIS Method. This approach uses information obtained from user interactions with the ARIS Toolset modeling environment and information from tool documentation. The application of the proposed approach results in language metamodels, clarifying the modeling elements of the ARIS Method and the relationships between these elements. The metamodels serve as a starting point for the definition of the semantics of the language and allow the construction of tools to manage modeling, simulation, analysis and transformation of organizational models and business process models.

## 1. INTRODUÇÃO

O ARIS Method (*ARChitecture for integrated Information Systems*) 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 [1].

Este método é estruturado em diferentes visões (organizacional, dados, controle, função e saída) e camadas de abstração [1]. Neste trabalho, nos concentramos na visão organizacional e na visão de função dos níveis mais altos de abstração. A *visão organizacional* descreve a hierarquia organizacional, isto é, descreve a comunicação e o relacionamento entre as unidades organizacionais de uma organização, revelando também os papéis dos indivíduos na organização. A *visão de função* 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. A sintaxe concentra-se puramente nos aspectos notacionais de uma linguagem, ignorando completamente os significados dos elementos sintáticos, que são revelados apenas através da semântica de uma linguagem. A definição semântica para uma linguagem  $L$ , ou simplesmente a semântica, é composta de duas partes: um *domínio semântico* e um *mapeamento semântico* dos elementos da sintaxe para o domínio semântico.

A Figura 1 apresenta a estrutura de uma linguagem de acordo com [2]. O mapeamento semântico é o vínculo entre cada elemento do conjunto sintático para algum elemento semântico do domínio da linguagem [2].

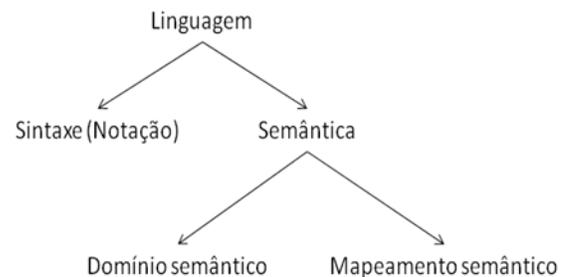


Figura 1. Estrutura de uma linguagem [2]

Com base nas diversas visões de linguagens de domínio presentes no ARIS Method e suas linguagens de modelagem, foram desenvolvidas várias ferramentas, como por exemplo, ARIS SOA Architect, ARIS Business Architect e ARIS Toolset. Estas ferramentas provêm um ambiente para modelagem, gerenciamento dos modelos produzidos e outros serviços de gerenciamento de processos [4]. Em particular, neste trabalho, estamos interessados nas linguagens de domínio presente no ARIS Method e seu suporte através do ARIS Toolset. O interesse nesta ferramenta deve-se à grande importância industrial desta ferramenta na prática de modelagem de processos de negócio e arquiteturas de TI.

Apesar de [1] apresentar a linguagem para representar cada uma das vistas da metodologia ARIS, a implementação das vistas no ARIS Toolset não condiz com o apresentado em [1]. Isso pode ser comprovado, pois há elementos de uma determinada linguagem de domínio (organizacional, por exemplo) que estão presentes na ferramenta de modelagem e não estão presentes nos modelos ou nas descrições textuais em [1] e vice-versa. Para exemplificar esta afirmação, no metamodelo organizacional apresentado em [1], existe o elemento Organizational Object e o

elemento Profile que não ocorre no ARIS Toolset, enquanto no ARIS Toolset apresenta o elemento System Organization Unit que não ocorre no metamodelo organizacional apresentado em [1].

Com essa deficiência na definição da linguagem, qualquer estudo, análise e construção de ferramentas baseada nas linguagens presentes no ARIS Method torna-se incompleta ou até mesmo difícil, pela ausência de documentação do conjunto completo e válido de elementos sintáticos e semânticos da linguagem. De fato, a fonte definitiva para estudar os elementos sintáticos e semânticos da linguagem é a implementação das ferramentas ARIS.

Como as ferramentas computacionais não relevam diretamente os elementos semânticos de uma linguagem, mas sim os elementos sintáticos concretos, a representação interna da linguagem do ponto de vista da ferramenta deve ser “escavada” na ausência de documentação definitiva para esta representação. Esta representação interna é chamada de *sintaxe abstrata* [2], que pode ser parcialmente definida através de um *metamodelo* [12].

Este trabalho tem por objetivo apresentar uma abordagem simples de escavação e definição das linguagens de domínio presentes no ARIS Toolset. Através das linguagens escavadas e bem definidas é possível esclarecer os elementos semânticos presentes em um determinado domínio do ARIS Method, no nosso caso o domínio organizacional (visão organizacional) e o domínio de processos de negócio (visão de função).

A seção 2 introduz a abordagem utilizada para escavação das linguagens de domínio presentes no ARIS Toolset e apresenta a aplicação da abordagem na escavação do domínio organizacional e do domínio de processos de negócio, a seção 3 apresenta uma discussão sobre a abordagem proposta neste artigo com os demais métodos utilizados para a escavação de modelos conceituais utilizado na atualidade e um debate sobre as diferenças entre os metamodelos de domínio escavados neste trabalho com os metamodelos propostos por Kern e Kühne [7] e aqueles apresentados por Scheer [1]. Por fim, a seção 4 apresenta as conclusões e os trabalhos futuros.

## 2. ABORDAGEM PROPOSTA

A abordagem proposta neste artigo tem por objetivo escavar e definir a linguagem de um determinado domínio identificando os elementos notacionais (sintáticos), os elementos que compõem a sintaxe abstrata (metamodelo) e a semântica de cada elemento que compõem o metamodelo e os relacionamentos entre os elementos do metamodelo.

Para realizar a escavação e construção da linguagem de um determinado domínio presente no ARIS Method como implementado no ARIS Toolset, as seguintes etapas são propostas:

- I. Identificar o modelo (diagrama) utilizado pelo ARIS Toolset para representar o domínio escolhido;
- II. A partir do modelo identificado na etapa anterior, identificar quais são os seus principais elementos. Essa identificação é importante, pois a partir dela é possível separar os elementos que são apenas notacionais (sintáticos) e desta forma, não pertencendo diretamente ao metamodelo do domínio do ARIS Method. Esse passo é necessário e importante porque ele determina quais são os elementos puramente notacionais e os

elementos que estão diretamente ligados a sintaxe abstrata da linguagem;

- III. Buscar informações na ferramenta que permitam esclarecer a semântica dos elementos selecionados na etapa (ii);
- IV. Realizar interações na ferramenta de modelagem com os elementos selecionados na etapa (ii) para revelar os relacionamentos entre estes elementos;
- V. Construir o metamodelo de domínio com as informações da etapa (ii), (iii), (iv);
- VI. Completar o metamodelo com especializações dos elementos identificados na etapa (ii).

As próximas subseções apresentam a construção de um fragmento da linguagem de domínio organizacional (seção 2.1) e da linguagem de processos de negócio (seção 2.2) como implementado no ARIS Toolset aplicando os passos I-VI desta abordagem.

## 2.1 Escavação e Construção do Metamodelo Organizacional do ARIS Method

### 2.1.1 Identificação do modelo de domínio

O ARIS Toolset utiliza o modelo Organizational Chart para representar as unidades organizacionais e seus inter-relacionamentos com os demais componentes da organização. O ambiente de modelagem disponibiliza os seguintes elementos para a modelagem em um Organizational Chart: Organization Unit Type, Organization Unit, Cost Center, Position Type, Position, System Organization Unit Type, System Organization Unit, Person Type, Internal Person, External Person, Group, Location, Workstation e Position Description.

### 2.1.2 Identificação dos principais elementos do modelo

O ARIS Toolset diferencia os elementos supracitados em duas categorias: (i) Object Types e (ii) Symbol Types. O primeiro conjunto representa um tipo abstrato de um elemento. O segundo conjunto representa as possíveis notações (símbolos gráficos) de um Object Type na ferramenta. Desta forma, um elemento de um determinado Object Type pode ser representado graficamente através de diversos símbolos e desta forma assumir diferentes significados de acordo com este símbolo. Por exemplo, o Object Type Person pode representar uma pessoa interna a organização (Internal Person) ou uma pessoa externa a organização (External Person) simplesmente mudando o seu Symbol Type. Podemos concluir que a especialização de tipos de objeto é feita através de tipos de símbolo para o mesmo tipo de objeto.

A Tabela 1 apresenta os Symbol Types que estão relacionados com os Object Types no diagrama Organizational Chart.

**Tabela 1 - Object Type e Symbol Type do Organization chart**

Object Type	Symbol Type
Organization Unit Type	Organization Unit Type
	Position Type
Organization Unit	Organization Unit
	Cost Center
Position	Position

System Organization Unit Type	System Organizational Unit Type
System Organization Unit	System Organizational Unit
Location	Location
	Workstation
Person	Internal Person
	External Person
Person Type	Person Type
	Position Description
Group	Group

Tomando como base neste passo apenas os Object Types identificamos as seguintes metaclasses do modelo organizacional do ARIS Method: Organization Unit Type, Organization Unit, Position, System Organization Unit Type, System Organization Unit, Location, Person, Person Type e Group. As especializações destas metaclasses (correspondentes aos Symbol Types) serão incorporadas no metamodelo apenas após a identificação das semântica destes elementos principais e da identificação dos relacionamentos entre eles.

### 2.1.3 Identificação da semântica dos elementos selecionados

Uma vez identificados os principais elementos, deve-se descrever o significado de cada elemento. A documentação online da ferramenta (*help*) da ferramenta foi utilizada como referência para este passo.

Um Organizational Chart baseia-se na identificação das unidades organizacionais (Organizational Unit) e seus inter-relacionamentos com os demais componentes da organização. Uma unidade organizacional é uma entidade responsável pela execução de atividades que materializem os objetivos da organização.

Os relacionamentos entre as unidades organizacionais, como as relações de superioridade e subordinação, são representadas no modelo na forma de relacionamentos (linhas que unem dois ou mais elementos de modelagem no ARIS Toolset) que podem assumir os seguintes significados: que uma unidade organizacional é tecnicamente superior a outra (*is technically superior to*); que uma unidade organizacional é disciplinarmente superior a outra (*is disciplinary superior to*) e que uma unidade organizacional é componente de outra unidade organizacional (*is a component of*).

O *help* do ARIS Toolset define que a menor unidade organizacional é representada por uma posição (Position) que um indivíduo (Person) possui dentro da organização. Desta forma, pode-se concluir que um Position é tipo (generalização) de Organization Unit. As posições podem ser alocadas de acordo com o tamanho das unidades, levando em consideração as regras do negócio ou a estrutura da empresa [1]. Várias posições podem ser associadas a uma unidade organizacional. As responsabilidades e deveres de uma posição (Position) são definidos no Position Description.

A associação de uma pessoa (Person) a uma unidade organizacional expressa que essa pessoa é um empregado da mesma e a associação de uma pessoa (Person) a uma posição (Position) define qual o status dela na organização, as funções e as responsabilidades dentro da organização.

As unidades organizacionais (Organizational Unit) e as pessoas (Person) podem ser agrupadas levando em consideração uma característica em comum como, por exemplo, as

responsabilidades e deveres que um conjunto de empregados ou que um conjunto de unidades organizacionais. Desta forma, o ARIS Method permite que sejam criados tipos de unidade organizacionais (Organizational Unit Type) como, por exemplo, departamentos e; tipos de pessoas (Person Type) como, por exemplo, gerente de departamento, líder de grupos ou gerente de projeto.

O benefício de utilizar os tipos (como, por exemplo, Organizational Unit Type e Person Type) é possibilidade de representar regras de negócios genéricas que são impostas a unidades organizacionais reais ou a pessoas da organização. Um exemplo disso é a possibilidade de especificar que somente um certo tipo de pessoa (Person Type) poderá desempenhar uma certa função ou ter acesso a uma determinada informação da organização.

Um grupo (Group) representa um grupo de empregados (Person) - com ou sem posições na organização - ou unidades organizacionais que estão trabalhando em conjunto por um período de tempo específico para um determinado objetivo.

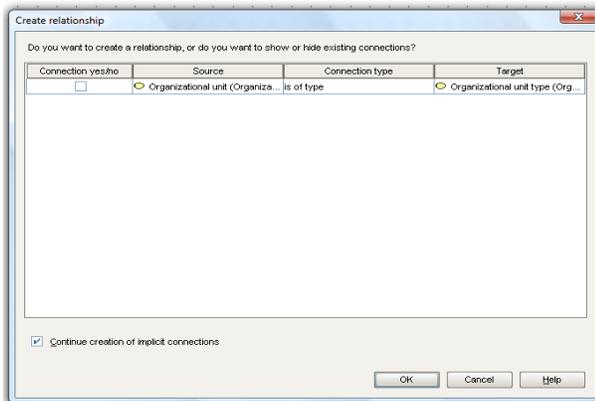
Location é um elemento de modelagem do ARIS que tem como finalidade representar a posição geográfica de uma unidade organizacional (Organizational Unit), pessoa (Person), posição (Position) ou de um recurso da organização. Uma localização (Location) pode representar uma região, uma cidade, uma planta de chão de fábrica, um prédio, uma sala ou uma estação de trabalho.

Para finalizar, de acordo com o *help* do ARIS Toolset, os sistemas de informação (System Organizational Unit) contêm estruturas organizacionais que foram levadas em consideração no momento da implantação dos sistemas. Dessa forma tais sistemas podem prover informação de como está organizada a estrutura organizacional da organização. A entidade System Organizational Unit Type representa a tipificação de um conjunto de System Organizational Units que possuem as mesmas características.

### 2.1.4 Identificação dos relacionamentos dos elementos selecionados

A quarta etapa da abordagem busca identificar os relacionamentos entre os elementos identificados na segunda etapa. Ao identificar os relacionamentos existentes entre elementos do modelo, é possível identificar conceitos que não foram devidamente explicitados ou até mesmo não apresentados no *help* da ferramenta.

A escavação dos relacionamentos se dá através da análise das informações presentes na interface e das informações apresentadas pela ferramenta na interação com o usuário durante a modelagem. A Figura 2 demonstra como a ferramenta trata os relacionamentos existentes entre dois elementos



**Figura 2. Demonstração de como os relacionamentos são explicitados na ferramenta.**

Ao tentar criar um relacionamento entre dois elementos de tipos distintos ou entre dois elementos de tipos iguais no ambiente de modelagem, como pode ser observado na Figura 2, a ferramenta de modelagem fornece quais são os relacionamentos possíveis entre os elementos.

A partir destas informações, é possível identificar outros relacionamentos entre os elementos do metamodelo organizacional e, no caso dos relacionamentos construídos a partir dos conceitos apresentados anteriormente, validá-los.

O ambiente de modelagem do ARIS Toolset (vide lado direito da Figura 2) especifica o seguinte padrão para identificar o elemento de origem, o nome do relacionamento, o elemento de destino do relacionamento: <ELEMENTO DE ORIGEM> - <NOME DO RELACIONAMENTO> - <ELEMENTO DE DESTINO>. Tomando como base esse padrão é possível extrair as seguintes informações do relacionamento apresentado na Figura 2:

- Elemento de origem: Organizational Unit;
- Nome do relacionamento: *is of type*;
- Elemento de destino: Organizational Unit Type;

Neste trabalho foi desenvolvida uma forma sistemática para a escavação dos relacionamentos entre os elementos presentes na sintaxe abstrata de um determinado domínio: **(a)** são determinados os auto-relacionamentos entre os elementos, como, por exemplo, os relacionamentos de elementos do tipo Organizational Unit com elementos do mesmo tipo; e **(b)** são identificados os relacionamentos entre os elementos de tipos diferentes, como, por exemplo, entre Organizational Unit e Organizational Unit Type.

Para exemplificar a forma sistemática de identificação dos relacionamentos dos elementos da sintaxe abstrata da linguagem que representa o domínio organizacional do ARIS Method implementada no ARIS Toolset, aplicamos a abordagem nos elementos Organizational Unit e Position.

Ao realizar a escavação dos auto-relacionamentos de Organizational Unit, foi possível identificar que o ARIS Toolset possui os seguintes conceitos expressos pelos relacionamentos encontrados: **a)** que uma unidade organizacional pode ser composta (*is composed of*) de outras unidades organizacionais; **b)** que uma unidade organizacional pode ter uma relação de superioridade hierárquica (*is superior*) em relação a outras; **c)** uma unidade organizacional pode ser tecnicamente superior (*is technical superior to*) ou disciplinarmente superior (*is disciplinary superior to*) a uma ou mais unidades da organização; e **d)** que uma

unidade organizacional pode ser responsável (*is responsible for*) por uma ou mais unidade organizacionais.

A idéia que uma unidade organizacional (Organizational Unit) é subordinada (*is subordinate*) a outra(s) unidade(s) organizacional(is) está implícita nos outros relacionamentos encontrados. Desta forma, ela não é considerada no modelo, pois não ocorre no ARIS.

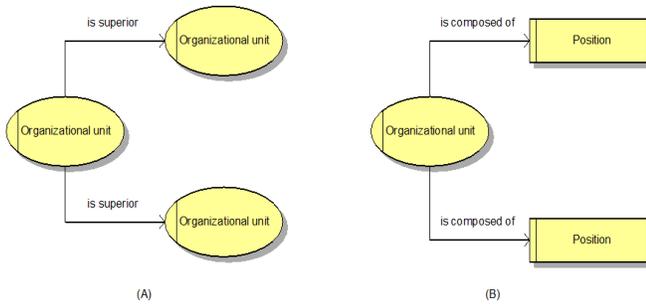
A escavação dos auto-relacionamentos do elemento Position revela os seguintes relacionamentos: **a)** uma posição pode ser substituída (*substitutes for*) para uma posição ou mais de uma posição; **b)** uma posição é tecnicamente superior (*is technical superior to*) ou disciplinarmente superior (*is disciplinary superior to*) a nenhuma, uma ou mais de uma posição e; **c)** uma posição é responsável por gerenciar (*is organization manager for*) nenhuma, uma ou mais de uma posição.

De acordo com a semântica apresentada na seção 2.3, uma posição (Position) deveria ter todos os auto-relacionamentos do elemento Organizational Unit (porque pela definição, uma posição é a menor unidade organizacional), mas a escavação dos auto-relacionamentos demonstrou que na sintaxe abstrata da linguagem do domínio organizacional do ARIS Toolset o elemento Position não possui todos os auto-relacionamentos do elemento Organizational Unit. Desta forma, conclui-se que não existe um relacionamento de generalização entre Position e Organizational Unit no nível do metamodelo e, portanto, o relacionamento de generalização proposto pela descrição presente no *help* da ferramenta não condiz com a real implementação na ferramenta de modelagem.

A segunda etapa da identificação dos relacionamentos é feita entre elementos de tipos diferentes como, por exemplo, a escavação dos relacionamentos entre Position e Organizational Unit.

Ao realizar a escavação dos relacionamentos entre Position e Organizational Unit, os seguintes foram identificados: **a)** uma unidade organizacional pode ser tecnicamente superior (*is technical superior to*) ou disciplinarmente superior (*is disciplinary superior to*) a uma ou mais posições (Position) e o inverso, ou seja, que uma posição (Position) pode ser tecnicamente superior (*is technical superior to*) ou disciplinarmente superior (*is disciplinary superior to*) a uma ou mais unidades organizacionais; **b)** uma unidade organizacional é composta por zero ou mais posições (Position); e **c)** uma pessoa com determinada posição (Position) na organização pode gerenciar (*is organization manager for*) uma ou mais unidade organizacionais.

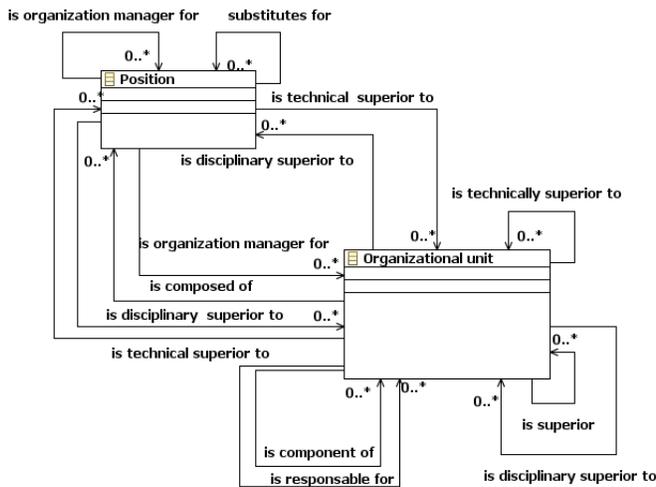
Tanto no caso **a)** quanto **b)** apresentados acima, é necessário identificar a cardinalidade de cada relacionamento. Para tal é necessário testar o relacionamento identificado com diversos elementos de destino e verificar se a ferramenta permite que um elemento de origem tenha zero ou mais relacionamentos com diferentes elementos de destino do mesmo tipo.



**Figura 3. Exemplo de relacionamentos entre elementos**

O relacionamento *is superior* apresentado entre a unidade organizacional na Figura 3 parte (A) representa os relacionamentos (no nível de metamodelo) que uma unidade organizacional (Organizational Unit) possui com outras unidades organizacionais. Como pode ser observado, o ARIS Toolset permite que esse elemento possa ter zero ou mais relacionamentos desse tipo e, assim, no metamodelo esse relacionamento possui cardinalidade 0..\*. A parte (B) da Figura 3 ilustra relacionamentos *is composed of* entre uma unidade organizacional (Organizational Unit) e uma posição (Position) possui cardinalidade 0..\*.

A partir dos relacionamentos escavados e dos conceitos apresentados foi possível construir uma metamodelo que representa a sintaxe abstrata da linguagem responsável pelo domínio organizacional implementado pelo ARIS Tools. A Figura 4 apresenta um fragmento da sintaxe abstrata da linguagem do domínio organizacional escavado pela abordagem proposta pelo artigo. Assim, a quarta e quinta etapa da abordagem proposta pelo artigo é concluída. O metamodelo foi definido através do Eclipse Modeling Framework (EMF).

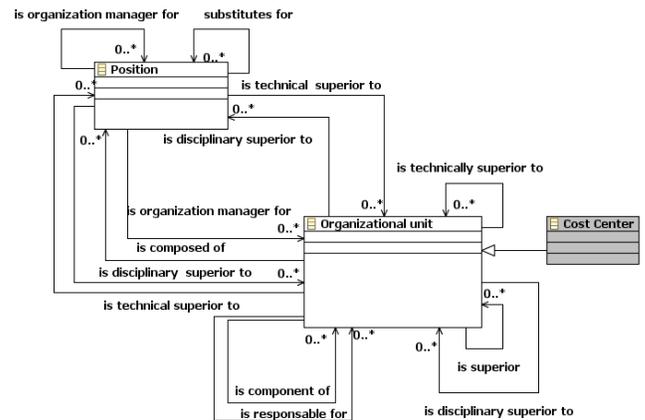


**Figura 4. Fragmento do metamodelo organizacional do ARIS escavado com base nas escavações dos relacionamentos.**

### 2.1.5 Adicionar elementos notacionais

A sexta, e última, etapa do desenvolvimento do metamodelo organizacional do ARIS corresponde à adição dos elementos que foram considerados notacionais na segunda etapa da abordagem proposta neste trabalho. Desta forma eles foram considerados apenas especializações dos elementos já presentes no metamodelo

construído. A Figura 5 apresenta um fragmento do metamodelo resultante da sexta etapa da abordagem.



**Figura 5. Fragmento da sintaxe abstrata da linguagem responsável pelo domínio organizacional com os elementos notacionais adicionados**

Como apresentado na Figure 5 o elemento Position não possui nenhum elemento notacional e o elemento Organizational Unit possui apenas o elemento notacional Cost Center (em cinza). Assim, o elemento Cost Center foi adicionado ao metamodelo como uma especialização da Organizational Unit.

Com a conclusão da última etapa da abordagem proposta neste artigo, temos um fragmento da linguagem que compõe o domínio organizacional do ARIS Method.

## 2.2 Escavação e Construção do Metamodelo de Processos do ARIS Method

Para a construção da linguagem do domínio de processos do ARIS Method foram executados os mesmos passos utilizados para a construção do metamodelo organizacional. As próximas subseções demonstraram outros aspectos da construção da linguagem como, por exemplo, a necessidade de adição de classes abstratas para melhor representação dos elementos.

### 2.2.1 Identificação do modelo de domínio

O ARIS Toolset utiliza o modelo eEPC (*Extended Event-Driven Process Chain*) para representar os processos organizacionais de uma organização. O ambiente de modelagem disponibiliza diversos elementos de modelagem para este modelo, para a demonstração da abordagem proposta foi selecionado um fragmento deste conjunto. Para uma melhor organização dos elementos de modelagem, o fragmento foi dividido nos seguintes conjuntos:

- I. Organizational: Organization Unit Type, Organization Unit, Cost Center, Position Type, Position, System Organization Unit Type, System Organization Unit, Person Type, Internal Person, External Person, Group, Location, Workstation, Position Description e Employee variable.
- II. Transformadores de informação: Function, Process interface, SAP business scenario, SAP business process, SAP function, SAP process configuration variant e Process module;
- III. Regras: AND rule, XOR rule, OR rule, Rule e Gateway.

IV. Eventos: Event, Start event, Intermediate event e End event.

### 2.2.2 Identificação dos principais elementos do modelo

A identificação dos principais elementos do modelo será feita da mesma forma que ocorreu na seção 2.1.2, ou seja, a diferenciação entre os elementos do fragmento entre Object Type e Symbol Type.

A Tabela 1 e a Tabela 2 apresentam os Symbol Types que estão relacionados com os Object Types no diagrama eEPC.

**Tabela 2. Object Type e Symbol Type do eEPC**

Conjunto	Object Type	Symbol Type
Organizacional	Employee variable	Employee variable
Transformadores de informação	Function	Function
		Process Interface
		SAP business scenario
		SAP business process
		SAP process configuration variant
		Process module
		SAP function
Regras	Rule	AND rule
		XOR rule
		OR rule
		Rule
		Gateway
Eventos	Event	Event
		Start event
		Intermediate event
		End event

O elemento Employee variable é o único elemento do conjunto organizacional que tem ocorrência somente no modelo eEPC. Os demais elementos organizacionais (Organization Unit Type, Organization Unit, Cost Center, Position Type, etc) ocorrem tanto no domínio organizacional quanto no domínio de processo.

### 2.2.3 Identificação da semântica dos elementos selecionados

Nesta seção, identificamos a semântica dos principais elementos selecionados a partir das informações contidas no help do ARIS Toolset.

Um evento (Event) representa um estado que é relevante para o gerenciamento do processo e que influencia ou controla o fluxo de execução de um ou mais processos de negócio. Mudanças no estado são refletidas na troca do status da(s) informação (ões) relevante(s) do processo. Eventos disparam atividades (Function) e são resultados de atividades. Um evento tem um aspecto temporal, ou seja, ocorre em um certo instante de tempo.

De acordo com o help do ARIS Toolset, uma atividade (Function) é uma tarefa técnica ou tarefa realizada sobre um determinado objeto, que possui como finalidade apoiar um ou vários objetivos de negócios da organização.

O elemento Rule representa os operadores lógicos que permitem especificar um relacionamento lógico entre eventos (Event) e atividade (Function) em processo de negócio.

O Employee variable representa um espaço reservado para uma pessoa que será especificada posteriormente, e cujo envolvimento no processo já pode ser identificado.

### 2.2.4 Identificação dos relacionamentos dos elementos selecionados

Esta seção seguirá os mesmos passos apresentados na seção 2.1.4, ou seja, primeiro iremos apresentar os auto-relacionamentos escavados dos elementos do fragmento do domínio de processos e depois os relacionamentos entre os elementos.

O elemento Function possui o auto-relacionamento denominado *is predecessor of*, que representa que uma atividade é predecessor de uma ou mais atividades.

O último auto-relacionamento encontrado na escavação pertence ao elemento regra (Rule). Este auto-relacionamento *links* é responsável por representar que uma regra pode possui um relacionamento com um ou mais regra.

Ao realizar a escavação dos relacionamentos entre Function e os elementos presentes no conjunto organizacional, percebeu-se que com exceção do elemento System Organization Unit, System Organization Unit Type e Location, os demais elementos possuem os mesmo relacionamentos com o elemento Function. Neste trabalho os elementos do conjunto organizacional com exceção do System Organization Unit, System Organization Unit Type e Location serão chamados de Actor.

De acordo com ARIS Toolset, um System Organization Unit pode estar associado (*is assigned to*) a nenhuma, uma ou mais atividades de um ou mais processo. Da mesma forma, o elemento System Organization Unit Type também pode estar associado (*can be assigned to*) a nenhuma, uma ou mais atividades de um ou mais processo. O ARIS Toolset também determina que uma atividade (Function) é realizada uma localidade, por causa disto, existe o relacionamento *is carried out* entre Location e Function.

Ao realizar a escavação dos relacionamentos entre Actor e Function, os seguintes foram identificados: **a)** O ator é responsável tecnicamente (*is technically responsible for*) por uma atividade; **b)** um ator realiza (*carries out*) uma atividade; **c)** O ator pode tomar decisões (*decides on*) na atividade; **d)** O ator contribui para (*contributes to*) a realização da atividade; **e)** O ator deve ser informado sobre (*must be informed about*) a atividade; **f)** O ator concorda ou aceita (*accepts*) realizar a atividade; **g)** O ator tem um papel de consultor (*has consulting role in*) na atividade; **h)** O ator deve ser informado sobre o cancelamento (*must be informed on cancellation*) da atividade e; **i)** O ator deve ser informado sobre o resultado (*must inform about result of*) da atividade.

A escavação demonstrou que o conceito apresentado na seção 2.2.3 que um evento dispara uma ou mais atividades e que uma atividade cria um ou mais eventos está presente na implementação da sintaxe abstrata da linguagem de processos no ARIS Toolset. Isto foi comprovado ao escavar os relacionamentos *creates* e *activates* entre Function e Event. Sendo o *creates* responsável por representar que uma atividade pode criar um ou mais eventos e o *activates* representando que um evento pode ativar uma ou mais atividades.

Segundo o ARIS Toolset, uma regra (Rule) ativar (*activates*) uma ou mais atividade do processo e uma atividade pode ser levar a (*leads to*) uma ou mais regras (Rule).

### 2.2.5 Construção do fragmento do metamodelo de processos

Esta seção apresentará diversos fragmentos da sintaxe abstrata da linguagem de processos do ARIS Method implementado no ARIS Toolset extraído a partir da abordagem proposta neste artigo.

A Figura 6 apresenta o fragmento da sintaxe abstrata de processos com os elementos organizacionais e seus relacionamentos com o elemento Function.

Com pode ser observado na Figura 6, o elemento Actor (em cinza) é uma classe abstrata no metamodelo que possui como finalidade de agrupar os relacionamentos, sem valor semântico para a linguagem de processos do ARIS Method.

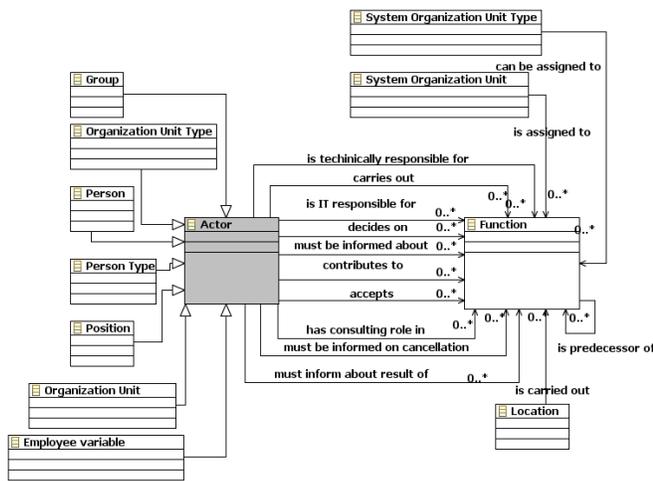


Figura 6. Fragmento da sintaxe abstrata da linguagem de processo com os elementos organizacionais e o elemento function.

Por fim, a Figura 7 apresenta o fragmento do metamodelo de processo com os elementos Function, Event e Rule e seus relacionamentos.

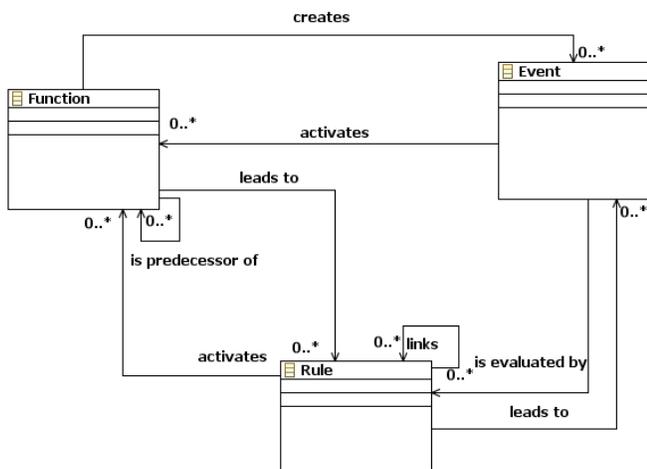


Figura 7. Fragmento do metamodelo de processo com os elementos Function, Event e Rule

### 2.2.6 Adicionar elementos notacionais

Com descrito na seção 2, sexta e última etapa da abordagem proposta neste artigo é a adição dos elementos considerados notacionais no metamodelo.

A Figura 8 apresenta uma exemplificação desta última etapa da abordagem. Nela foram adicionados apenas os elementos notacionais (em cinza) correspondentes ao conjunto organizacional presente no domínio de processos no ARIS Method.

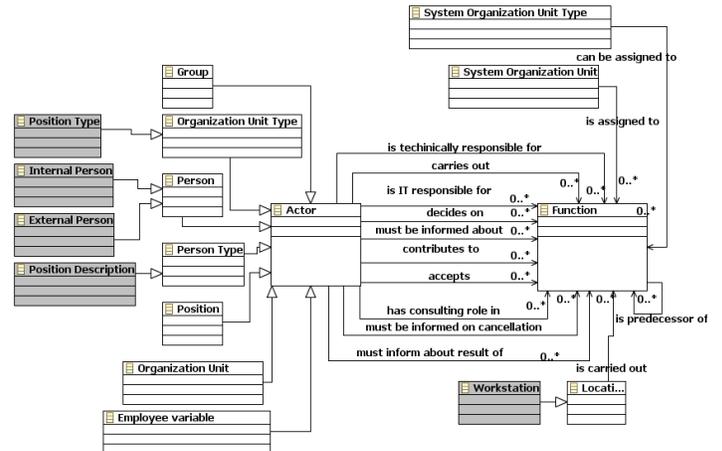


Figura 8. Fragmento do metamodelo de processo do ARIS Method escavado com alguns elementos notacionais adicionados

## 3. DISCUSSÃO

Diversas técnicas de construção de *modelos conceituais* a partir da extração de informações de aplicativos têm sido desenvolvidas [3], a maior parte delas extraíndo informações a partir de componentes do software como, por exemplo, bancos de dados [4], esquemas XML [5] e interfaces gráficas [6]. Diferentemente destas técnicas, a abordagem de escavação proposta neste trabalho tem por objetivo a escavação e definição da *linguagem de modelagem* de um determinado domínio que está presente dentro de uma ferramenta.

Como pode ser observado na Figura 9, Kern [7] propôs um metamodelo para o repositório do ARIS Toolset que não diferencia os diferentes tipos de objetos do ARIS Method, como, por exemplo, unidades organizacionais (*Organizational Units*), atividades (*Functions*) e eventos (*Event*) [1]. Estes tipos são codificados em identificadores dentro dos elementos como, por exemplo, *modelType*, *typeNum* e *cxnDefType* nos respectivos elementos *Model*, *ObjDef* e *CxnDef*. Como consequência não é possível através de uma inspeção rápida do modelo diferenciar tipos que representam categorias conceituais completamente diferentes, como, por exemplo, unidades organizacionais e atividades.

Os metamodelos apresentados neste artigo possuem um grau de abstração maior que o proposto pelo Kern. Isso pode ser comprovado porque os diferentes tipos de objetos são definidos explicitamente como metaclasses.

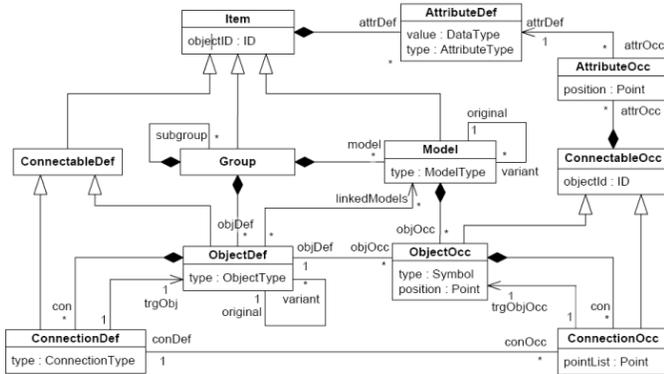


Figura 9. Estrutura de repositórios do ARIS [7]

Para finalizar a discussão, a sintaxe abstrata do domínio organizacional e a sintaxe abstrata do domínio de processo propostos neste artigo, possuem maior fidelidade com as reais sintaxes abstratas implementadas na ferramenta ARIS Toolset do que as apresentadas em [2]. Assim sendo, os metamodelos apresentados neste artigo, são mais ricos pra análise, estudo e construção de ferramentas.

## 4. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, propomos uma abordagem de escavação dos metamodelos das linguagens de domínio presentes no ARIS Method. O interesse nesta ferramenta deve-se à sua grande importância na prática de modelagem de processos de negócio e arquiteturas de TI.

A abordagem utiliza informações obtidas a partir das interações do usuário com o ambiente de modelagem e assim como as informações contidas no *help* da ferramenta ARIS Toolset. A ferramenta foi considerada como a referência definitiva, na ausência de documentação detalhada e atualizada.

Diferentemente dos trabalhos apresentados em [4], [5] e [6], o trabalho apresentado neste artigo visa escavar uma linguagem de modelagem, relevando tanto a sintaxe abstrata quanto uma descrição informal da semântica de cada elemento desta linguagem. Em relação ao trabalho de Kern e Kühne [7], este artigo apresenta um conjunto de metamodelos que explicita os diferentes tipos de objetos (unidade organizacionais, atividades e eventos) presentes no ARIS Method.

Os metamodelos identificados neste trabalho viabilizam a construção de ferramentas de modelagem, simulação e análise de modelos de processos de negócio usando técnicas de desenvolvimento orientado a modelos (*Model-Driven Design*). Os metamodelos propostos neste artigo viabilizam ainda a construção de ferramentas que implementem transformações entre linguagens distintas em um nível de abstração alto sem a necessidade de realizar transformações em nível de documento, como é o caso da transformação XSLT apresentada em [8] para a conversão de eEPCs do ARIS [1] para EPML (EPC Markup Language) [8].

Por fim, em trabalhos futuros utilizaremos ontologias de fundamentação como, por exemplo, a UFO [9], [10] para permitir uma definição rigorosa da semântica do metamodelos do ARIS Method. A técnica de definição da semântica nos permitirá também identificar elementos de modelagem inadequados de cada domínio através de uma série de critérios sistemáticos de avaliação revelando a clareza, expressividade, completude,

parsimônia, lucidez, correção e consistência das linguagens de modelagem [11], levando à definição de recomendações de melhoria bem fundamentadas.

## 5. AGRADECIMENTOS

Este trabalho foi desenvolvido com recursos da Fundação de Apoio à Ciência e Tecnologia do Espírito Santo (FAPES) no escopo do projeto INFRA-MODELA e com recursos do Fundo de Apoio à Ciência e Tecnologia do Município de Vitória (FACITEC) no escopo do projeto MODELA.

## 6. REFERÊNCIAS

- [1] Scheer, A. W., Scheer, 2000. "ARIS-Business Process". 3a edição, Springer.
- [2] Harel, D. and Rumpe, B. 2000 *Modeling Languages: Syntax, Semantics and all that Stuff, Part I: the Basic Stuff*. Technical Report. UMI Order Number: MCS00-16., Weizmann Science Press of Israel.
- [3] Felhberg, G. 2007. *Interoperabilidade Semântica de Aplicações Comercialmente Disponíveis Usando Ontologias de Domínio*. Monografia de Conclusão de Curso, Universidade Federal do Espírito Santo.
- [4] Volz, R., Oberle, D., Steffen, S. and Rudi, S. 2002. OntoLiFT Prototype - WonderWeb: Ontology Infrastructure for the Semantic Web.
- [5] Ferdinand, M., Zirpins, C. and Trastour, D. 2004. Lifting XML Schema to OWL, Web Engineering - 4th International Conference, ICWE 2004, LNCS 3140, Springer, 26-30.
- [6] Hsi, I. 2005. *Analyzing the Conceptual Integrity of Computing Applications Through Ontological. Excavation and Analysis, Dissertação de Doutorado*, Georgia Institute of Technology.
- [7] Kern, H. and Kühne, S. 2007. Model Interchange between ARIS and Eclipse EMF, *The 7th OOPSLA Workshop Domain-Specific Modeling*, Computer Science and Information System Reports, Technical Reports, TR-38, University of Jyväskylä, Finland.
- [8] Mendling, J. and Nüttgen, M. 2004. Transformation of ARIS Markup Language to EPML. *Proc. of the 3rd GI Workshop on Event-Driven Process Chains*.
- [9] Guizzardi, G. 2005. *Ontological Foundations for Structural Conceptual Models*. PhD with Cum Laude. Telematica Instituut Fundamental Research Series, vol. 015. Enschede, the Netherlands: Telematica Instituut, 2005.
- [10] Guizzardi, G., Vascelos, J., Segrini, B., Falbo, R., and Guizzardi, R. 2008. Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology, *IDEAS, XI workshop Iberoamericano*, Brasil, Pernambuco, Fevereiro.
- [11] 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 MODELS Conference*, LNCS vol. 3713, Springer-Verlag.
- [12] Object Management Group, 2003, MDA-Guide, V1.0.1, omg/03-06-01.