



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

A UTILIZAÇÃO DE PADRÕES DE PROJETO DE ONTOLOGIAS NA
MODELAGEM DE UM CENÁRIO REAL

KAREN NUNES VALCARCEL

Orientador
FERNANDA ARAUJO BAIÃO AMORIM

RIO DE JANEIRO, RJ – BRASIL
OUTUBRO DE 2014

A UTILIZAÇÃO DE PADRÕES DE PROJETO DE ONTOLOGIAS NA
MODELAGEM DE UM CENÁRIO REAL

KAREN NUNES VALCARCEL

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovada por:

FERNANDA ARAUJO BAIÃO AMORIM (UNIRIO)

KATE CERQUEIRA REVOREDO(UNIRIO)

FLAVIA MARIA SANTORO(UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

OUTUBRO DE 2014

Dedico esta monografia aos meus amores, sempre presentes. Obrigada pelo incentivo e pelo apoio de vocês.

Agradecimentos

A Deus e ao meu São Jorge guerreiro que me iluminaram e protegeram, me levando e trazendo em segurança todos os dias dessa jornada. Ao meu pai, Ronaldo Valcarcel, que me proibiu de desistir quando eu achei que não conseguiria e que me incentivou em todos os momentos me guiando sempre da melhor maneira possível, me ajudando em todas as minhas escolhas. A minha mãe, Maria Edith Nunes Valcarcel, com todo seu apoio sempre me ajudando de todas as formas que podia, e tornando as coisas mais fáceis. Vocês foram fundamentais para eu chegar até aqui, sempre presando pelo meu futuro. Obrigada por todo esforço e sacrifício.

Ao meu noivo, Daniel de Miranda Chaves Christiani, que passou de meu melhor amigo nesta jornada, chamada Sistema de Informações, e vem se tornando meu companheiro pra toda a vida, por toda a paciência, amizade e cumplicidade. A minha orientadora, Fernanda Araújo Baião Amorim, sempre presente me mostrando qual o próximo passo para seguir em meu projeto, esclarecendo minhas dúvidas com paciência nas nossas reuniões que muitas vezes se estenderam pela Ponte Rio-Niterói.

Obrigada a todos, essa vitória é nossa.

RESUMO

Modelar é representar conceitualmente um cenário, descrevendo suas características estruturais a fim de facilitar o seu entendimento. Através da modelagem podemos representar até mesmo os cenários mais complexos, reutilizando soluções já conhecidas, chamadas de Padrões de Projeto. Estas soluções estão disponíveis e catalogadas em repositórios de Padrões de Projeto de Ontologia (ODP – *Ontology Design Patterns*). Neste trabalho foi criado um modelo conceitual da ontologia de um Departamento Pessoal através da adaptação e reutilização de padrões de projeto de ontologia disponíveis em um repositório de padrões de projeto de ontologia online, alimentado com os padrões criados à partir das experiências de outros usuários de Ontologias. Neste processo de criação da Ontologia do cenário proposto foram demonstradas as vantagens e desvantagens encontradas durante a instanciação desses padrões ao cenário real modelado, bem como as facilidades asseguradas em relação à criação de um modelo sem o uso desses padrões de ontologias, como, por exemplo, a facilidade em se familiarizar com os conceitos ou as dificuldades em se conhecer e se entender o cenário que está sendo modelado. O foco deste trabalho foi a análise dos benefícios e limitações do uso de padrões de projeto de ontologia pré-existentes na modelagem de um cenário real, e as vantagens obtidas neste processo em relação à qualidade e expressividade.

Palavras-chave: Ontologias, Padrões de Ontologias, Repositórios de ODPs, Modelagem, Cenário Real.

ABSTRACT

Modeling is the act of conceptually representing a scenario describing their structural characteristics in order to facilitate its understanding. Through modeling we can represent even the most complex scenarios, reusing already known solutions, called Design Patterns. These solutions are available and cataloged in Ontology Design Patterns (ODP) repositories.

In this work a conceptual model of the ontology of a Personnel Department was created through the reuse of Ontology Design Patterns available on a repository supplied with the patterns created from specialists. While creating the Ontology in the proposed scenario, this work shows the advantages and disadvantages encountered to adjust existing ODPs to the real scenario modeled, as well as the facilities obtained when creating the model compared to the non-ODP scenario. For example, we experienced the facility brought by ODPs to become familiar with the domain concepts and the difficulties in knowing and understanding the scenario being modeled. The focus of this work is the representation of the suitability of pre-existing ODP to a real scenario, and the advantages gained in this process regarding the quality and expressiveness.

Keywords: Ontology, Ontology Patterns, ODP Repositories, Modeling, Real Scenario

Índice

1	Introdução	10
1.1	Motivação.....	10
1.2	Problema	11
2	Fundamentação Teórica	12
2.1	Ontologias	12
2.2	Vantagens em se trabalhar com ontologias.....	15
2.3	Linguagens de Representação de modelos.....	15
2.3.1	RDF (Resource Description Framework)	16
2.3.2	OWL - Ontology Web Language.....	16
2.3.3	UML - Unified Modeling Language.....	18
2.4	Ferramentas utilizadas.....	19
3	Padrões de Projeto de Ontologias	20
3.1	Uso e Benefícios	21
3.2	Tipos de Padrões	21
3.3	Utilizando Padrões de Projeto.....	24
3.4	Criação de Padrões de Projeto de Ontologias	30
3.5	Repositórios de Padrões de Ontologias.....	32
4	Modelagem do cenário trabalhado.....	34
4.1	Apresentação do cenário modelado	34
4.2	Modelagem do cenário de testes	36
4.3	Criação do cenário sem utilização de Padrões de Projeto.....	37
4.4	Análise da modelagem sem uso de padrões	38
4.5	Criação do cenário usando Padrões de Ontologias	39
4.6	Análise sobre o uso de padrões de Ontologias.....	47
5	Conclusão e Trabalhos Futuros.....	50

Índice de Tabelas

Tabela 1: Padrão <i>Person</i>	25
Tabela 2: Padrão <i>ParticipantRole</i>	26
Tabela 3: Padrão <i>TimeInterval</i>	27
Tabela 4: Padrão <i>Participation</i>	28
Tabela 5: Padrão <i>Place</i>	29
Tabela 6: Padrão <i>Set</i>	30
Tabela 7: Modelando Casos de uso - Do Genérico ao Específico [Gangemi e Presutti, 2009].....	32

Índice de Figuras

Figura 1: Ontologia sobre conceito de Automóvel.....	13
Figura 2: Tipos de Ontologia por [Guarino, 1997].....	14
Figura 3: Arquitetura de Representação do Conhecimento [Martins, 2007].....	16
Figura 4: Expressividade da Linguagem OWL [Autor desconhecido]	17
Figura 5: Tipos de padrões de projeto de Ontologias [Gangemi, 2004].	24
Figura 6: Modelagem do Cenário de testes	36
Figura 7: Janela <i>DataProperty</i> do Padrão <i>TimeInterval</i> no Protege	38
Figura 8: Padrões para as classes Funcionário - Servidor – Contratado	40
Figura 9: Comparativo entre o padrão <i>Person</i> e seu uso visto no Protegé	40
Figura 10: Padrões para as classes Função – Servidor	41
Figura 11: Comparativo entre o padrão <i>ParticipantRole</i> e seu uso visto no Protegé	42
Figura 12: Padrões para representar as ausências de um servidor.....	42
Figura 13: Comparativo entre o padrão <i>TimeInterval</i> e seu uso no Protegé	43
Figura 14: Padrão para representar o Pagamento ou Auxílio recebido pelo servidor	44
Figura 15: Comparativo entre o padrão <i>Participation</i> e seu uso no Protegé.....	44
Figura 16: Padrão para as classes Servidor – Notas	45
Figura 17: Padrão <i>Place</i> para definir os locais de Publicações.....	46
Figura 18: Cenário Proposto para o uso de Padrões	47

1 Introdução

Neste capítulo serão apresentados a motivação deste trabalho e o problema que iremos tratar.

1.1 Motivação

A motivação deste projeto vem da necessidade de um estudo sobre a importância da utilização de padrões de projeto de ontologias disponíveis em repositórios próprios, e sua importância em se trabalhar a reutilização e o compartilhamento desses padrões de ontologias. O uso de repositórios de padrões de projeto de ontologias facilita o acesso aos padrões de ontologias pré-definidos uma vez que serve como base de consulta às soluções criadas anteriormente a partir das experiências de outros usuários de ontologias. Para entender melhor o tema, é necessário entender a origem destes termos.

Primeiramente explicamos o significado de Ontologia. O “Estudo do ser” tem origem do grego, onde *ontos* significa "ser", e *logos*, "saber", "doutrina". De origem filosófica, a palavra ontologia refere-se, segundo Gruber [1993], a “uma especificação explícita de uma conceitualização”, ou seja, ao modelo de dados através do qual um conjunto de conceitos é representado dentro de um domínio e quais os relacionamentos serão possibilitados pelas correlações dos mesmos.

Definida a ideia de ontologias podemos expor a ideia dos padrões de projeto de Ontologias (ODP), ou simplesmente padrões de ontologias, nos quais são estabelecidas as diretrizes comuns a fim de solucionar problemas de projeto de ontologia, ou seja, um ODP pode ser definido como uma solução de modelagem para resolver um problema recorrente[Gangemi,2004].

A importância de se utilizar um repositório de padrões de ontologias está no fato de que os padrões de projetos já criados, melhorados e catalogados podem ser mais bem

reutilizados quando agrupados em um repositório, uma vez que estes repositórios representam uma infraestrutura criada para facilitar este o acesso aos padrões elaborados por usuários de ontologias que colaboram não só disponibilizando novos padrões, mas também atualizando e comentando os já existentes.

A utilização de padrões de projeto de ontologias não é simples. Primeiro é necessário entender a ideia geral deste padrão e a abrangência que ele pode atender, e até mesmo cada conceito envolvido no modelo, assim como os relacionamentos envolvidos, pois um padrão é na verdade um conjunto de conceitos que juntos formam uma situação que se deseja representar. Muitas vezes pode-se deixar levar pelo nome do padrão ou pela ideia de um desses conceitos que estão envolvidos, porém deve-se atentar se ele realmente se encaixa no cenário que está sendo modelado.

Esse trabalho visa ilustrar como a reutilização de padrões de projeto de ontologia pode auxiliar a modelagem de um cenário real de uma organização. Foi feita uma comparação das ontologias construídas com e sem o uso dos ODPs obtidos de um repositório de padrões de ontologia existente na literatura.

1.2 Problema

O problema abordado nessa proposta será uma análise mais aprofundada dos benefícios da utilização dos padrões de projeto de ontologias disponíveis para uso em um repositório próprio traçando uma comparação entre criar uma modelagem utilizando padrões e outra sem a utilização dos mesmos com o propósito de clarear os conhecimentos existentes sobre o assunto.

2 Fundamentação Teórica

2.1 Ontologias

Conforme mencionado anteriormente, a palavra ontologia refere-se ao modelo de dados através do qual um conjunto de conceitos é representado dentro de um domínio e quais os relacionamentos serão possibilitados pelas correlações dos mesmos.

Em Gruber [1993] encontramos uma definição de ontologia como “uma especificação explícita de uma conceituação”, onde é possível entender que uma ontologia pode ser definida como um conjunto de termos de representação no qual nomes de entidades são associáveis a uma interpretação lógica no universo do discurso. Formalmente, uma ontologia é a afirmação de uma teoria lógica, através da qual diferentes campos podem ser explorados, como por exemplo, inteligência artificial, através do raciocínio dedutivo; representação do conhecimento de um domínio específico, com o relacionamento semântico; e padronização de comunicação através da reutilização de informações semânticas para se aumentar a produção de novas ontologias, uma vez que incentivando os desenvolvedores, na investigação e compreensão dos modelos existentes, os resultados podem ser mais favoráveis do que quando é necessário o desenvolvimento das ontologias desde a sua criação. De Guarino [2004] podemos entender que ontologias computacionais são o meio para modelar formalmente a estrutura de **um domínio (Sistema)**, ou seja, explicitar quais são **os conceitos (as entidades)** e as relações existentes entre estes conceitos, que são relevantes para um propósito.

Para esta formalização alguns componentes foram definidos em Gruber [1993]:

- **Classes:** maneira de se representar qualquer coisa em um no domínio de interesse, como por exemplo, uma tarefa, ou uma função, ou uma estratégia, etc.
- **Relações:** representação de um tipo de interação entre os conceitos no domínio, ou seja, entre as classes e os elementos do domínio;

- **Axiomas:** representam as sentenças que são sempre verdadeiras, independente da situação, aplicadas na definição de restrições sobre relações e classes. Um axioma é uma afirmação lógica;

- **Instâncias:** representam os conceitos e relacionamentos considerando os elementos do domínio.

Um exemplo prático destes conceitos pode ser visto na ontologia automóvel, representada na Figura 1. O termo automóvel refere-se à classe/entidade “automóvel”, pois o mesmo tem propriedades que o definem (rodas, motor...). Esta classe se relaciona com as demais através das relações, que podem ou não ser hierárquicas, como por exemplo, triciclo é um automóvel. Para tratar de um triciclo específico você pode instanciar-lo, “meuTriciclo”, por exemplo. Caso exista interesse em se fazer uma afirmação (um triciclo é um automóvel que possui 3 rodas), usa-se um axioma.

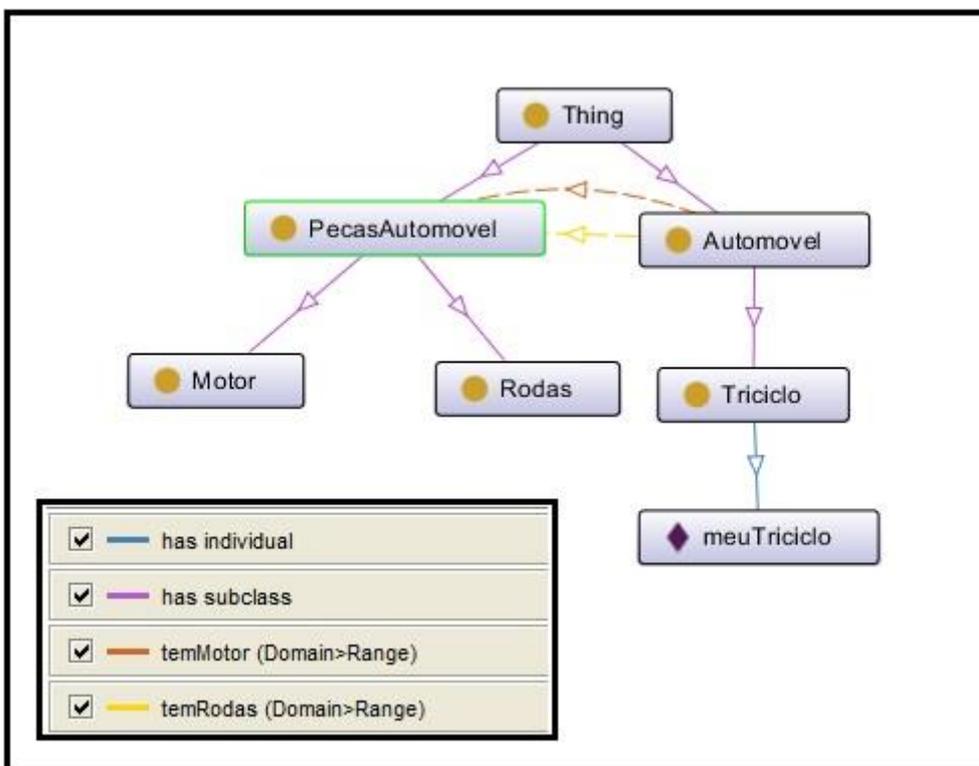


Figura 1: Ontologia sobre conceito de Automóvel

No trabalho de Guarino [1997] ontologias recebem diferentes classificações:

- **Ontologias de alto nível:** grupo o qual inclui o vocabulário relacionado a conceitos globais, não pertencendo a um único domínio. Esses conceitos mais gerais são genéricos e podem ser especializados por conceitos existentes nas ontologias de domínio ou tarefa.
- **Ontologias de domínio:** englobam ontologias que representam um determinado domínio, podendo ser reutilizáveis dentro do mesmo. São representados seus conceitos e relacionamentos, as atividades relacionadas, teorias e princípios desse domínio.
- **Ontologia de tarefas:** englobam conjuntos de conceitos, atividades e recursos utilizados para solucionar problemas (tarefas) recorrentes a diversos domínios distintos. Dessa forma, estão inclusos nomes de atividades genéricas e informações necessárias para o seu detalhamento, podendo ser reutilizadas em diferentes domínios.
- **Ontologias de aplicações:** compõe o grupo formado pelas ontologias criadas a partir de conceitos ou de uma tarefa específica aplicada a um domínio específico.

Esses tipos de ontologias podem ser entendidos graficamente através da Figura 2.

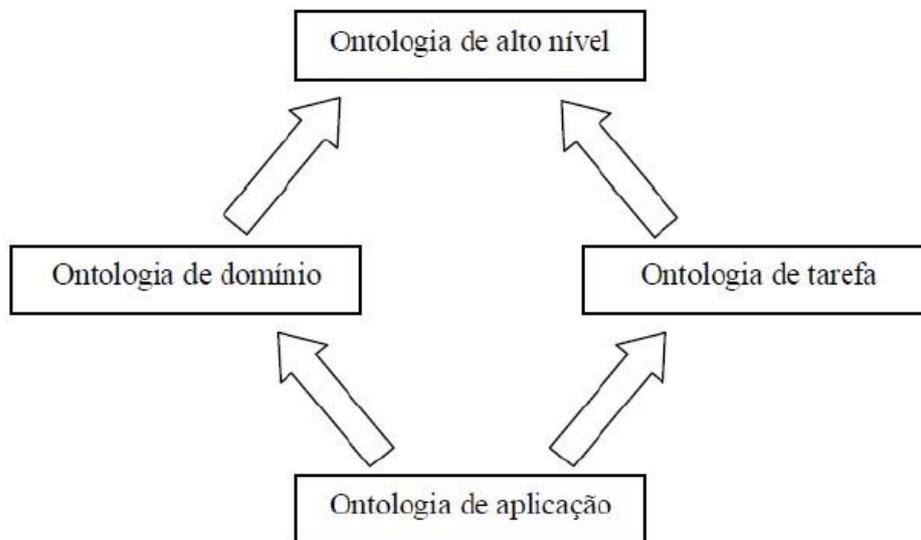


Figura 2: Tipos de Ontologia por [Guarino, 1997]

2.2 Vantagens em se trabalhar com ontologias

Conforme visto em [Ceci, 2010] a aplicação de ontologias no contexto de um determinado domínio proporciona uma maior organização e, com isso, facilita a reutilização da informação. Assim, a compreensão deste domínio pode ser compartilhada entre pessoas e sistemas, acrescentando estruturas semânticas a uma fonte de dados, para facilitar uma troca de informações desses dados.

Moraes [2007] cita em seu trabalho uma lista de vantagens de se trabalhar com ontologias nas diversas áreas da computação, entre elas a Recuperação de Informações na Internet, por exemplo, no caso de catálogos de produtos online ou mesmo através de ferramentas de análise para que seja realizada uma pesquisa contextualizada, o Processamento de Linguagem Natural para facilitar o entendimento, por exemplo, em ambientes corporativos realizando armazenamento de informações, entre outras vantagens.

Moody [2003] define um conjunto de características importantes que refletem a qualidade semântica de um modelo conceitual, que podemos considerar de forma análoga ao tratar de ontologias como a representação de um modelo conceitual. Dentre estas características, destacam-se a completude (preenche todos os requisitos do usuário?), simplicidade (mínimo possível de classes e relacionamentos), flexibilidade (facilidade em lidar com alterações), integração (consistência do modelo de dados com o restante dos dados), além das referentes à compreensão (facilidade com que os conceitos e estruturas de dados podem ser compreendidos) e execução (facilidade de ser implementada dentro dos requisitos existentes).

No trabalho de [Castro, 2010] também é definido que as linguagens de modelagem de menor expressividade são aquelas de aprendizado e uso mais simples. Ao longo do projeto serão abordadas outras vantagens de se trabalhar com ontologias.

2.3 Linguagens de Representação de modelos

Com o passar dos anos, diversas linguagens foram criadas para representar ontologias. Segundo Sell [2006] podemos extrair ontologias a partir de documentos XML (*eXtensible Markup Language*). Essas ontologias são representadas por diferentes

linguagens. Algumas das linguagens mais utilizadas (Figura 3) são a RDF (*Resource Description Framework*) e a OWL (*Ontology Web Language*), que serão descritas abaixo.

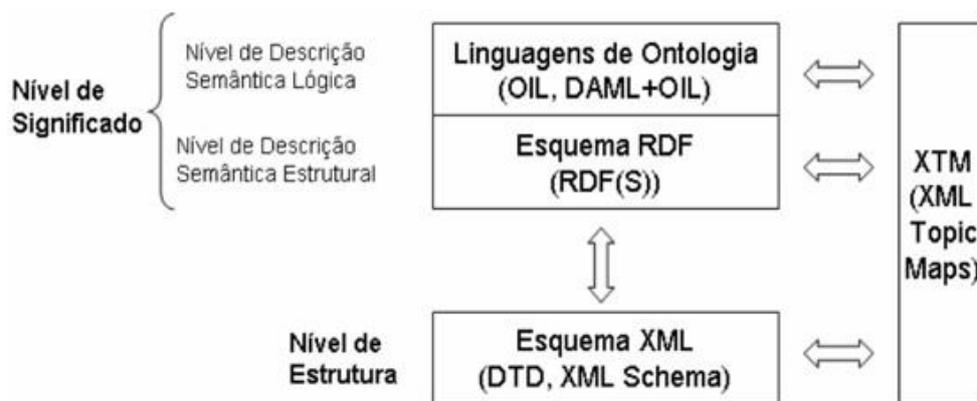


Figura 3: Arquitetura de Representação do Conhecimento [Martins, 2007]

2.3.1 RDF (Resource Description Framework)

Segundo Sell [2006] o modelo RDF é um modelo de dados baseado em triplas (objeto, predicado e valor), através das quais são permitidas as descrições sobre conteúdo independentemente da estrutura do mesmo. O RDF Schema (W3C, 2004) compreende as primitivas para a organização de hierarquias e para a definição de restrições sobre RDF.

Por Fensel, [2001] e Gomez-Perez,[2006], citados por Sell [2006], entendemos que o RDF *Schema* é definido como um formalismo para a estruturação de ontologias mais simples. No caso de se trabalhar com inferências, são necessários formalismos mais expressivos para a representação de ontologias, pois estes devem incluir suporte para axiomas e outras formas de relacionamentos entre os conceitos [Vieira et al, 2005].

2.3.2 OWL - Ontology Web Language

A OWL é uma linguagem de representação e armazenamento de ontologias, criada com o objetivo de definir e instanciar ontologias, e garantindo-lhes um maior poder de expressividade que outras linguagens por possuir um vocabulário mais rico do que as linguagens anteriores, o que enriquece sua representação. Em uma ontologia representada em OWL são incluídas descrições de classes, propriedades e suas instâncias. A OWL foi baseada em OIL e DAML + OIL sendo atualmente esta

linguagem a recomendada da W3C [W3C, 2004], a fim de definir e instanciar ontologias na web, se subdividido em três sublinguagens: OWL *Lite*, OWL DL e OWL *Full*, que se diferenciam por suas particularidades abaixo descritas como visto em Lopes [2009] e em Antoniou e van Harmelen [2004] em relação a sua expressividade e computabilidade [Azevedo *et al.*, 2008] conforme mostrado na Figura 4.

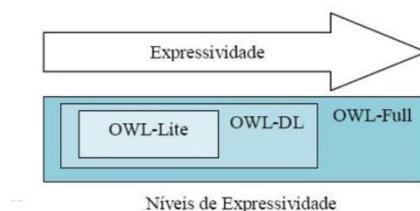


Figura 4: Expressividade da Linguagem OWL [Autor desconhecido]

- **OWL *Lite*:** Sublinguagem com menor poder descritivo, caracterizada pelos construtos mais simples da linguagem, como classe, subclasse, propriedade de tipo de dado e objeto, sub-propriedade, domínio, alcance, indivíduos, semelhanças entre classes, propriedades e indivíduos, diferenças entre indivíduos, propriedades inversas, transitivas, simétricas, funcionais, inverso funcionais, “todos os valores de”, “alguns valores de”, interseção de classes, com restrições de cardinalidade entre zero e um para propriedades, porém sendo mais fácil de entender e de se implementar;
- **OWL DL (*Description Logics*, ou Lógicas de Descrição):** formada pelo conjunto completo de construtos, além de restrições de cardinalidade completas, operações de conjunto (tais como união, complemento, interseção e classes enumeradas), disjunção, axiomas de intervalo de valor para propriedade de dados, restrição de um conjunto de instâncias específicas para uma propriedade (*allValuesFrom*, *someValuesFrom*), definindo classes com essa restrição, e regra de classificação de uma instância caso possua um valor específico em uma propriedade (*hasValue*), características estas que não estão presentes na OWL *Lite*. Porém nesta sublinguagem não é possível a classificação como uma instância ou a criação de uma instância como um classe, permitindo que seja manipulável por uma máquina em um tempo determinado;

- **OWL Full:** da mesma forma que a OWL DL, possui o conjunto completo de construtos OWL, além de permitir que uma classe seja identificada como uma instância, assim como permite que uma instância seja identificada como uma classe, impedindo que seu tratamento computacional tenha um tempo definido.

Segundo Azevedo *et al.* [2008] o modelo OWL provê um conjunto de vocabulário mais rico do que o encontrado no modelo RDF para melhor restringir o conjunto de triplas que podem ser representada.

Segundo [Vieira et al, 2005] o RDF é uma linguagem de meta-dados genérica usada para explicitar relacionamentos entre recursos. O RDF permite especificar o que são recursos, o que são relacionamentos e quais são os relacionamentos entre os recursos em um domínio qualquer. A OWL é uma evolução do RDF onde definições mais detalhadas sobre as relações podem ser expressas através de ontologias. [Vieira et al, 2005] também cita que através da utilização da OWL é possível descrever características especiais sobre conceitos e os relacionamentos, através de axiomas lógicos.

2.3.3 UML - Unified Modeling Language

Outra linguagem utilizada na modelagem conceitual de cenários é a UML, uma linguagem para especificação, visualização, construção e documentação de artefatos de sistema de software, assim como para modelos de processo e sistemas que não são softwares [Engiel, 2011], ou seja, para se representar de forma gráfica um cenário .

Entre os benefícios de se utilizar a UML para modelar ontologias, Cranefield et al, [1999] cita que a linguagem UML possui uma comunidade de usuários muito grande e rápida expansão, e, portanto seus usuários serão mais propensos a se familiarizar com este notação; além disso, a linguagem UML permite uma representação gráfica padrão, importante para permitir que os usuários de sistemas de informação distribuídos possam procurar uma ontologia e descobrir os conceitos que podem aparecer em suas consultas. Como a linguagem UML permite que sejam definidos diversos tipos de diagrama que podem ser usados para modelar um comportamento dinâmico de um sistema estático, essa linguagem foi escolhida para representar os diagramas utilizados neste trabalho, ou seja, não só para representar o cenário modelado, mas também o cenário modelado através do uso de padrões.

2.4 Ferramentas utilizadas

Ao longo do projeto algumas ferramentas foram usadas para ilustrar conceitos e cenários. São elas:

- **Protégé**

Segundo a Wiki disponibilizada pelo site ¹desenvolvedor do Software, o Protégé é uma plataforma de código aberto que fornece uma comunidade de usuários em crescimento, com um conjunto de ferramentas para a construção de modelos de domínio e aplicações baseadas no conhecimento com ontologias. Neste projeto foi utilizado o Protégé 4.3 para a criação das ontologias do cenário de testes. Através desse Software é possível criar as entidades e os relacionamentos entre eles, na linguagem ontológica escolhida, seja ela RDF ou OWL.

- **AstaH**

Neste projeto foi utilizado o *Astah Community 6.7.0/43495* como uma ferramenta voltada para a modelagem de sistemas utilizando a UML. Os modelos do cenário abordado neste projeto foram desenhados com esta ferramenta. Através dela também é possível representar diagramas de caso de uso (anexo), e diagramas de classe e diagramas entidade-relacionamento, que não foram abordados para este projeto.

¹ http://protegewiki.stanford.edu/wiki/Main_Page

3 Padrões de Projeto de Ontologias

Nos anos 70, Christopher Alexander lançou a ideia de Padrões de Projeto, através de seu trabalho no campo da Arquitetura [Gangemi, 2004]. Esta ideia se baseia na subdivisão de áreas de conhecimento baseando-se em problemas comuns vistos anteriormente e nas soluções utilizadas. Este trabalho impulsionou a ideia entre a comunidade de software, primeiramente no que trata a orientação objeto, a publicar um livro sobre Padrões de Projeto nesta área [Gama *et al.*, 1997].

Gangemi *et al* [2007] introduz seu trabalho dizendo que um padrão de projeto, na Engenharia de Software, é uma maneira simples e elegante para solucionar problemas específicos de modelagem orientada a objeto oferecendo um vocabulário comum para auxiliar na comunicação, documentação e exploração de diferentes modelos, ou seja é uma descrição genérica de como resolver a modelagem de um problema que futuramente poderá ser utilizada em diversas situações. Um Padrão de Projeto reflete a experiência, o conhecimento e as ideias dos desenvolvedores que tiveram sucesso usando esses padrões em seus trabalhos, criando soluções adaptáveis a diferentes problemas. Esses padrões de projeto podem estar organizados em catálogos os quais são subdivididos em diferentes categorias, utilizando a linguagem OWL e a notação UML para descrever as ontologias.

No campo de Ontologias, o termo Padrões de Projeto de Ontologias (*Ontology Design Patterns* - ODP), segundo o trabalho de Gangemi [2004], introduz as referências às diretrizes comuns com o objetivo de solucionar problemas de projeto de ontologia, ou seja, um ODP pode ser definido como uma solução de modelagem para resolver um problema recorrente. Já em Gangemi [2005] podemos acrescentar a definição de que ODPs são soluções de modelo iniciais para problemas específicos em um determinado cenário. Desta forma, podemos considerar que as ODPs são ferramentas adicionais para ajudar a realizar tarefas tais como aprimorar as ontologias, por exemplo.

Há algum tempo que os Padrões de Projeto de Ontologia (ODP) surgiram como uma forma de ajudar desenvolvedores de ontologias para modelar projetos utilizando a linguagem OWL. No entanto, um problema na utilização desses ODPs são as dificuldades de se entender a linguagem pelos praticantes que não estão familiarizados

com a sintaxe OWL, mas que estão habituados a modelar usando outras técnicas de engenharia de software. Desta maneira, uma notação gráfica (por exemplo, UML) ajudaria os engenheiros de software e usuários de ontologias na tarefa de entender os padrões, e aumentaria a reutilização de Padrões de Projeto de Ontologias no processo de construção de ontologias.

3.1 Uso e Benefícios

Segundo Aranguren [2005] e Guedes [2013], um Padrão de Projeto (*Design Pattern*) pode ser definido como abstrações de soluções para os problemas de modelagem que ocorrem com determinada frequência em diferentes ontologias. Através desses padrões pode-se criar a documentação dessas soluções para um problema de projeto em um determinado cenário.

A vantagem de usar Padrões de Projeto de Ontologia é a ajuda obtida para produzir ontologias com melhores estruturas cujo conhecimento vem com maior facilidade. Outra vantagem é que usando estes padrões, as ontologias tornaram-se mais expressivas, ou seja, uma maior quantidade de informações acerca dos conceitos envolvidos é agregado ao modelo projetado [Aranguren, 2005].

Atualmente, com a grande procura pela reutilização de ontologias é importante tomar algumas precauções. Ao utilizar padrões para desenvolver ODPs podemos, de maneira mais simples, evitar erros de modelagem, além de aperfeiçoar as ontologias, e a manutenção e reutilização das mesmas como visto em Egaña [2008], e Mortensen [2012].

3.2 Tipos de Padrões

Gangemi [2004] e Presutti [2008] agrupam os ODPs em seis diferentes famílias que se subdividem de acordo com suas especificidades, conforme mostrado na Figura 5, e também descrito abaixo, ou seja, grupos orientados para diferentes tipos de usuários para tratar de diferentes problemas.

- **ODPs Estruturais (*Structural ODPs*):**

- **ODPs Lógicos (*Logical ODPs*)** são ontologias genéricas compostas por construções lógicas com o objetivo de resolver um problema de expressividade. Por serem expressas em termos de vocabulário lógico possuem assinaturas vazias. Essas ODPs são independentes da existência de um domínio específico de interesse, ou seja, independentes de um conteúdo. Outra observação importante é que um mesmo ODP pode ser utilizado mais de uma vez em um mesmo projeto de ontologia [Presutti, 2008]. Um exemplo de aplicação de ODPs lógicos é na representação de uma relação entre mais do que dois elementos, com o objetivo de expressar uma relação semântica *n*-ária somente usando classe e relação binária primitivas.
- **ODPs de Arquitetura (*Architectural ODPs*)** são formadas por grupos de ODPs lógicas, com o objetivo de formar um *design* geral para a ontologia. Estas ODPs surgiram a partir de necessidades específicas de restrições computacionais complexas. São especialmente úteis em referências de documentação [Gangemi, 2004].
- **ODPs de Correspondência (*Correspondente ODPs*):**
 - **ODPs de Reengenharia (*Re-engineering ODPs*)** são regras de transformação aplicadas de modo a criar uma nova ontologia (modelo-alvo) a partir de elementos de um modelo fonte, ou seja, transformar um modelo conceitual. Pode-se subdividi-las em dois tipos: reengenharia de esquema e de refatoração.
 - **ODPs de Mapeamento (*Mapping ODPs*)** referenciam a possibilidade semântica de comparação entre duas ontologias mapeáveis existentes. Existem três relações semânticas básicas que são utilizadas para as afirmações de mapeamento: equivalência, contenção, e sobreposição. O foco dessas ODPs é fazer a relação de duas ontologias, sem alterar os tipos lógicos dos elementos de ontologias envolvidos. Um exemplo de ODP de Mapeamento é utilizar a propriedade “*equivalentTo*” para comparar ontologias de mesmo significado, por exemplo, caso houvesse

a necessidade de fazer a comparação entre as ontologias Servidor e Contratado modeladas neste trabalho.

- **ODPs de Conteúdo** (*Content ODPs*) se relacionam com a codificação de conceitos, e não com os padrões lógicos do projeto, fornecendo soluções para problemas de modelagem de domínio e afetando unicamente áreas específicas da ontologia tratando, assim, dos problemas de modelagem deste domínio. É comum a reutilização dos mesmos. Os exemplos de ODPs de conteúdo foram utilizados na modelagem por padrões de projeto no cenário teste deste trabalho.
- **ODPs de Raciocínio** (*Reasoning ODPs*) são normatizações, ou seja, são aplicações de ODPs lógicas a fim de se obter resultados determinados de raciocínio. Alguns exemplos de ODPs de raciocínio são aquelas utilizadas para realizar normatizações, como por exemplo, para instanciar uma classificação, ou uma subordinação.
- **ODPs de Apresentação** (*Presentation ODPs*)
 - **ODPs de Nomeclatura** (*Naming ODPs*) são convenções para se criar nomes dentro dos *namespaces* de acordo com boas práticas para o bom entendimento pelos desenvolvedores dos padrões de ontologias.
 - **ODPs de Anotação** (*Annotation ODPs*) fornecem as propriedades ou os esquemas de anotação que podem ser utilizados a fim de melhorar a inteligibilidade das ontologias e os seus elementos.
- **ODPs Léxico-Sintáticas** (*Lexico-Syntactic ODPs*) refere-se às estruturas linguísticas ou esquemas os quais consistem em que determinadas palavras sigam uma ordem específica, permitindo a generalização e a extração de algumas conclusões sobre seus conceitos, sendo útil para realizar associações simples entre lógica e conteúdo, ou seja, podem ser utilizadas para propósitos didáticos.

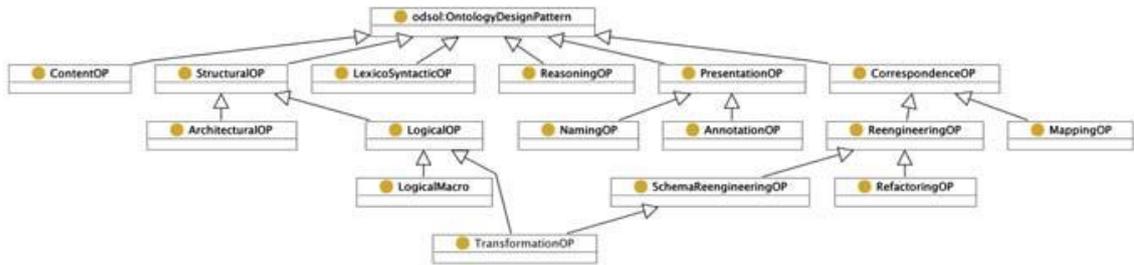


Figura 5: Tipos de padrões de projeto de Ontologias [Gangemi, 2004].

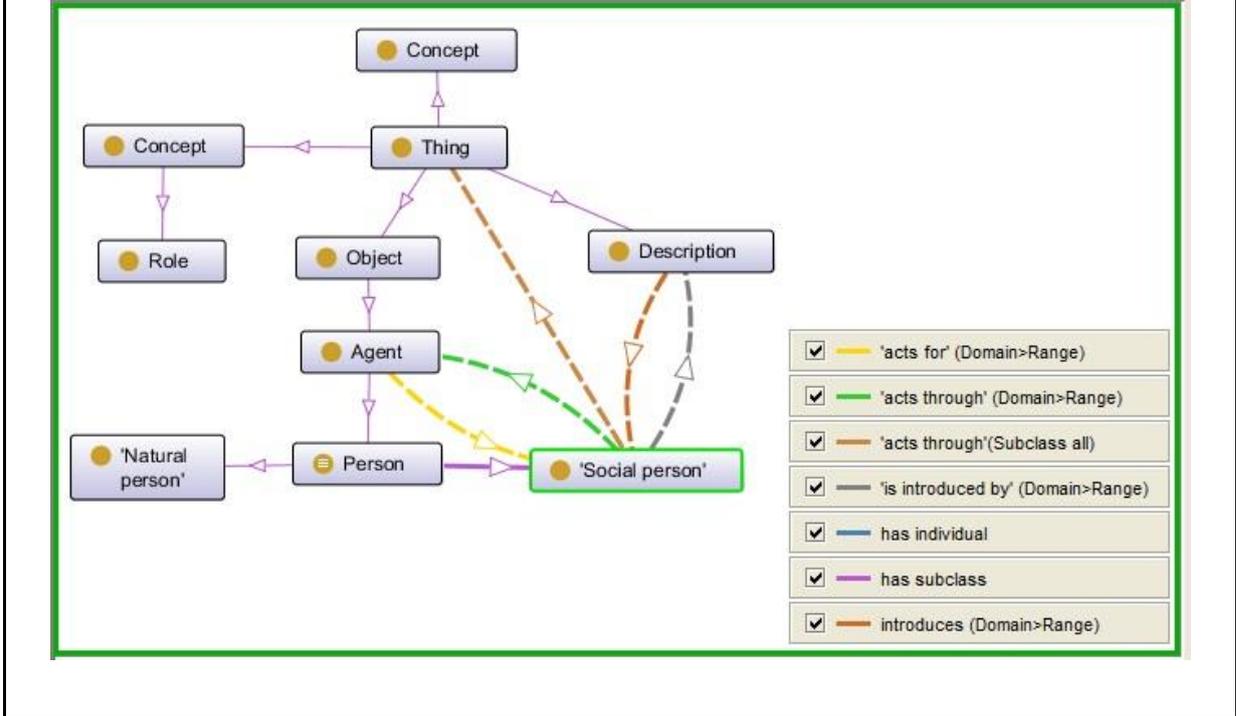
3.3 Utilizando Padrões de Projeto

Uma vez conhecido o significado de um ODP e seus benefícios, podemos, nesta seção, apresentar os OPDs de conteúdo disponíveis para consulta e utilização no repositório de ODPs escolhido como base para este trabalho, no site <http://ontologydesignpatterns.org/>.

Os ODPs aqui descritos serão trabalhados neste projeto em um próximo capítulo.

Padrão Person

Representação do padrão no Protégé

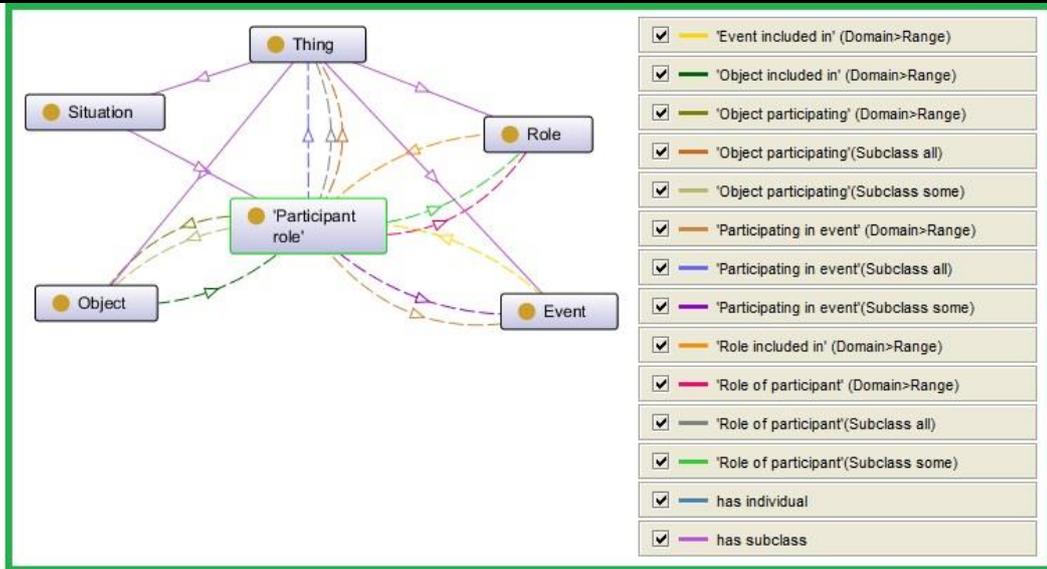


Descrição	Exemplos de uso
<p>Padrão utilizado para representar pessoas no seu estado natural ou social. Considera o indivíduo (“<i>Natural Person</i>”) como ser atuante seja ao representar um papel ou ao realizar uma tarefa (“<i>Social Person</i>”). A pessoa, nesse contexto equivale a um objeto (<i>Object</i>) atuando (<i>Agent</i>) em um papel, uma tarefa específica (<i>Role</i>) que pode ser descrita (<i>Concept</i>).</p>	<p>Funcionário, Servidor, Contratado.</p>

Tabela 1: Padrão *Person*

Padrão *ParticipantRole*

Representação do padrão no Protégé



Descrição

Este padrão tem como objetivo representar os participantes e o papel que desempenham (*ParticipantRole*) em um evento (*Event*) num dado momento (*Situation*). Ele desconsidera qualquer aspecto relacionado ao tempo na participação. Um evento é qualquer processo físico, social ou mental, evento ou estado. Um objeto é qualquer objeto físico, social ou mental, ou substância. Um papel é um conceito que classifica o objeto. E esta classe é uma situação que representa o papel de um objeto específico participando de um evento.

Exemplos de uso

Cargo de Chefia

Tabela 2: Padrão *ParticipantRole*

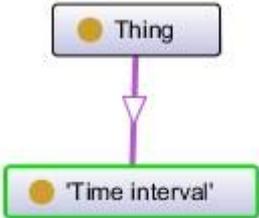
Padrão <i>TimeInterval</i>	
Representação do padrão no Protégé	
	
Descrição	Exemplos de uso
Padrão utilizado quando o objetivo é representar qualquer intervalo temporal, com data de início e data de fim, independente da existência de qualquer evento.	Ausência (Férias, Licença, ou Falta), e Frequência Integral

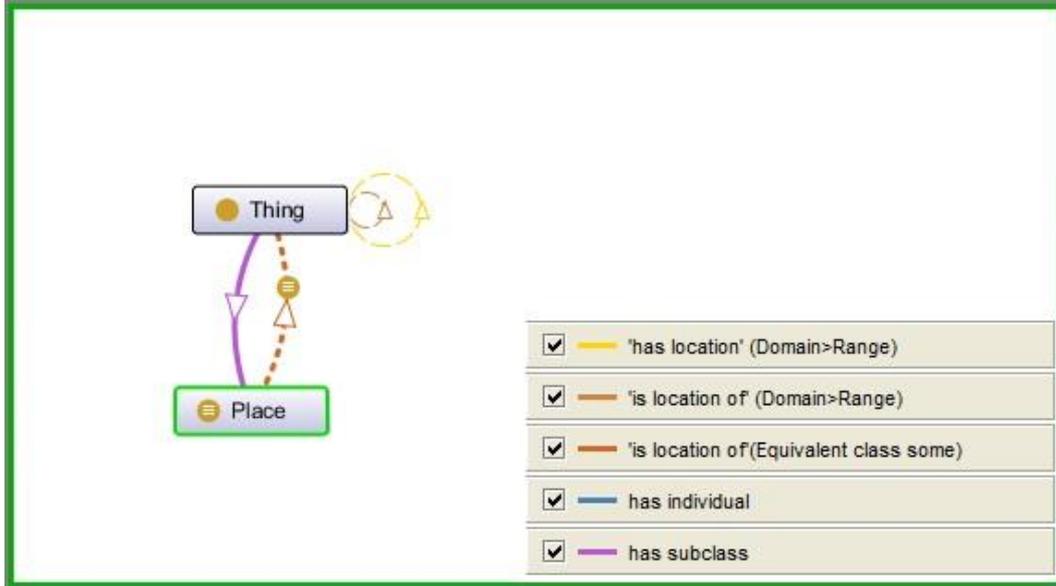
Tabela 3: Padrão *TimeInterval*

Padrão <i>Participation</i>	
Representação do padrão no Protégé	
Descrição	Exemplos de uso
<p>Padrão que representa a participação de um objeto em um evento. Permite-nos, usando restrições de cardinalidade, definir o número de participantes de um evento. Um objeto é qualquer objeto físico, social ou mental, ou substância. E um evento é qualquer processo físico, social ou mental, evento ou estado.</p>	<p>Pagamento.</p>

Tabela 4: Padrão *Participation*

Padrão *Place*

Representação do padrão no Protégé



Descrição

Padrão utilizado para representar locais, lugares de coisas, exemplo: Um veículo de publicação. Referencia uma localização, seja ela parcial, física, social ou metafórica.

Exemplos de uso

Diário Oficial,
Boletim de
Serviço.

Tabela 5: Padrão *Place*

Padrão Set	
Representação do padrão no Protégé	
<pre> classDiagram Thing < -- Collection Collection < -- set </pre>	
Descrição	Exemplos de uso
Padrão de representação de conjuntos de coisas. Um conjunto é uma coleção que não pode conter elementos duplicados. Uma coleção representa o conceito de um conjunto de entidades (coisas) que compartilham uma ou mais propriedades.	Avaliação de Desempenho e o conjunto de notas

Tabela 6: Padrão Set

3.4 Criação de Padrões de Projeto de Ontologias

Segundo Gangemi [2005], o primeiro passo para se trabalhar com ODPs é dominar a noção de "Casos de Uso Genéricos", ou seja, como realizar a generalização de casos de uso que foram fornecidos como exemplos para uma questão de modelagem de domínio.

O objetivo de se trabalhar com casos de uso é modelar uma expressão de um tema recorrente em diferentes projetos, independentemente de qual linguagem lógica especial foi adotada. Neste projeto o cenário de testes será trabalhado de duas maneiras, criando os padrões de projeto, e reutilizando padrões já existentes.

O autor citou algumas perguntas frequentemente utilizadas para se modelar um projeto de ontologia, que são as seguintes:

- Quem faz o quê, quando e onde?

- Que objetos participam de um determinado evento?
- Quais são as partes de uma coisa?
- Do que é feito um objeto?
- Qual é o lugar de alguma coisa?
- Qual é o período de tempo de alguma coisa?
- Qual a técnica, o método, a prática está sendo usada?
- Quais as tarefas devem ser executadas para atingir um determinado objetivo?
- Este comportamento está em conformidade com uma determinada regra?
- Qual é a função desse artefato?
- Como é construído este objeto?
- Qual é a concepção de que o artefato?
- Como esse fenômeno aconteceu?
- Qual é o seu papel nessa transação?
- Sobre o que essa informação é? Como é percebida?
- Qual modelo de argumentação que você está adotando para a negociação de um acordo?
- Qual é o intervalo de confiança que você dá a este axioma?

No anexo deste trabalho estão alguns exemplos de casos de uso para o cenário modelado, demonstrando algumas das questões as quais deveriam ser respondidas para que fosse possível a criação de casos de usos genéricos podemos ver pela Tabela 7 como adaptar essas questões para casos de usos específicos, o que será visto no Capítulo 5, quando estivermos criando o nosso cenário de testes.

Generic Competency Questions	Specific Modelling Use Case
Who does <u>what</u> , <u>when</u> and <u>where</u> ?	Production reports, schedules
Which objects <u>take part in</u> a certain event?	Resource allocation, biochemical pathways
What are the <u>parts</u> of something?	Component schemas, warehouse management
What's an object <u>made of</u> ?	Drug and food composition, e.g. for safety (comp.)
What's the <u>place</u> of something?	Geographic systems, resource allocation
What's the <u>time</u> frame of something?	Dynamic knowledge bases
What <u>technique, method, practice</u> is being used?	Instructions, enterprise know-how database
Which <u>tasks</u> should be <u>executed</u> in order to achieve a certain goal?	Planning, workflow management
Does this behaviour <u>conform</u> to a certain rule?	Control systems, legal reasoning services
What's the <u>function</u> of that artifact?	System description
How is that object <u>built</u> ?	Control systems, quality check
What's the <u>design</u> of that artifact?	Project assistants, catalogues
How did that phenomenon <u>happen</u> ?	Diagnostic systems, physical models
What's your <u>role</u> in that transaction?	Activity diagrams, planning, organizational models
What that information <u>is about</u> ? How is it <u>realized</u> ?	Information and content modelling, computational models, subject directories
What <u>argumentation model</u> are you adopting for negotiating an agreement?	Cooperation systems
What's the <u>degree of confidence</u> that you give to this axiom?	Ontology engineering tools

Tabela 7: Modelando Casos de uso - Do Genérico ao Específico [Gangemi e Presutti, 2009]

3.5 Repositórios de Padrões de Ontologias

Como foi dito anteriormente, repositórios de padrões de ontologias facilitam na modelagem de um cenário ajudando a manter a qualidade da ontologia que esta sendo desenvolvida, uma vez que os padrões disponíveis para utilização no repositório foram modelados por usuários de ontologias baseando-se em suas próprias experiências, ou mesmo foram aprimorados pela comunidade, discutidos e eventualmente aprimorados por seus membros.

Hashemi [2009] Aprofunda o conhecimento acerca dos repositórios de uma forma cujo entendimento é simplificado ao dizer que um repositório de ontologias funciona como um "mapa" das teorias conhecidas, enquanto Grüninger [2010] se aprofunda nesse cenário ao representar o uso de ontologias dentro de um repositório como uma relação entre informações semânticas, por ele chamadas de subteorias, e classifica as teorias dentro do repositório como módulos.

Supondo que já existem os conceitos das ontologias em um determinado repositório podemos, a partir destes, construir novos modelos de teorias, ou seja, adaptar ou instanciar os padrões existentes para reutilizar a informação disponível. No caso dos

repositórios de padrões de ontologia a ideia é a mesma: o repositório de ODPs utilizado neste trabalho fornece um catálogo de padrões divididos de acordo com os possíveis tipos de padrões, apresentando-os pelos seus nomes, objetivos, domínio e usuários.

4 Modelagem do cenário trabalhado

Este capítulo visa a criação do modelo conceitual de um cenário real, como base para a criação de um modelo ontológico dos conceitos envolvidos no funcionamento de um Departamento Pessoal de uma empresa pública brasileira. Nesse estudo foi utilizado o repositório de padrões de ontologias disponível no site *www.ontologydesignpatterns.org*. Os padrões escolhidos para esta modelagem foram de conteúdo. A escolha dos padrões a serem instanciados requereu uma análise minuciosa da descrição, objetivos e domínio dos padrões documentados no repositório, e da sua representação em OWL, que foi importada na ferramenta Protégé para visualização. Esta análise dos ODPs buscou avaliar a adequabilidade de cada um deles para solucionar situações específicas no cenário modelado.

Para a modelagem de qualquer domínio, é necessário determinar a abrangência desejada. Neste trabalho, essa escolha foi feita por ser um cenário o qual não possuía uma ontologia e cujo conhecimento era dominado pela autora deste trabalho.

Definido o escopo do cenário, neste estudo de caso foram criadas duas ontologias que o representassem: uma reutilizando os padrões de projeto de ontologias existentes e a outra sem a reutilização de padrões. A partir daí, foi possível comparar as ontologias geradas quanto à representação de cada conceito envolvido no cenário, desde suas características e atributos até mesmo a possibilidade de sugestão de melhorias aos processos realizados.

4.1 Apresentação do cenário modelado

O Departamento de pessoal HL é composto por vários setores organizados de acordo com suas funções e destina-se a resolver as diversas situações e solicitações dos funcionários da mesma. Os funcionários podem ser de dois tipos: Servidor (quando o mesmo entra mediante a realização de um concurso público) e Contratado (quando o funcionário realiza um contrato junto ao órgão governamental que rege a empresa). Todos os servidores possuem uma lotação em um departamento, e recebem um Salário Base.

A cada dia, registra-se a presença ou ausência de um servidor ao trabalho na empresa HL. Chama-se de Frequência Integral a presença diária, no período de um mês do servidor no Trabalho. Caso o servidor não tenha comparecido todos os dias, ele terá esta ausência registrada. Estas ausências podem se classificadas como Férias (mês de descanso após o servidor completar um ano trabalhado), falta (ausência injustificada por um numero específico de dias), ou licença (médica por motivo de saúde, etc...) de diferentes prazos identificadas por um código.

O salário Base é composto de um valor de salário bruto e de gratificações individuais de cada servidor. Além disso, o servidor também poderá receber um valor adicional ao seu salário base, isto é um auxílio. Dentre os auxílios existentes estão os referentes a saúde, transporte e alimentação oferecidos a todos os servidores, e os auxílios exclusivos para casos específicos como auxílio natalidade e o auxílio pré-escolar, por exemplo, quando os servidores têm filhos. O auxílio transporte tem um valor variável de acordo com o seu endereço e o regime de trabalho (plantão ou diarista). O auxílio alimentação tem um valor fixo para todos os servidores, independente do cargo. Outros auxílios disponíveis são o de adicional noturno, e dependendo do departamento que esteja alocado, auxílios de insalubridade ou periculosidade, definidos de acordo com o local e as condições de trabalho a que são submetidos, acordo com o laudo médico realizado por um perito.

Alguns servidores da empresa HL podem receber uma determinada função (por exemplo, de chefia) em um determinado período, após ser designado para esse cargo.

O servidor pode procurar o Departamento Pessoal para solicitar diferentes requisições, sendo que algumas requisições podem gerar os auxílios já mencionados, e outras requisições somente uma alteração nos dados cadastrados do servidor. Cada solicitação de um servidor é encaminhada a outro departamento da empresa (Protocolo), e se tornará um processo. Um processo é composto pela solicitação, os documentos necessários e uma resposta à requisição. Ao ser finalizado, um processo gera uma publicação, que poderá ser feita em dois diferentes meios: Diário Oficial ou Boletim de serviço. Cada processo é arquivado na pasta cadastral do servidor.

Os servidores recebem, anualmente, avaliações de desempenho, na qual lhe são atribuídos dois tipos de notas. Uma referente a uma auto avaliação, outra atribuída por

seu superior (ou por um deles quando o servidor possuir mais de uma chefia). Esta nota gera um valor de gratificação a ser pago no salário base do servidor.

O funcionário contratado é administrado por uma empresa externa, portanto não é o foco deste trabalho, embora seja mencionado neste modelo.

4.2 Modelagem do cenário de testes

Utilizando a ferramenta *Astah Community 6.7.0/43495*, foi criada uma modelagem do cenário em UML baseando-se nesta descrição textual do cenário proposto para análise (minimundo) estruturando o funcionamento do mesmo, ou seja, demonstrar através da modelagem conceitual quais são as atividades envolvidas no Sistema e a maneira que estas se relacionam, criando um modelo do seu estado atual, como pode ser observado na Figura 6.

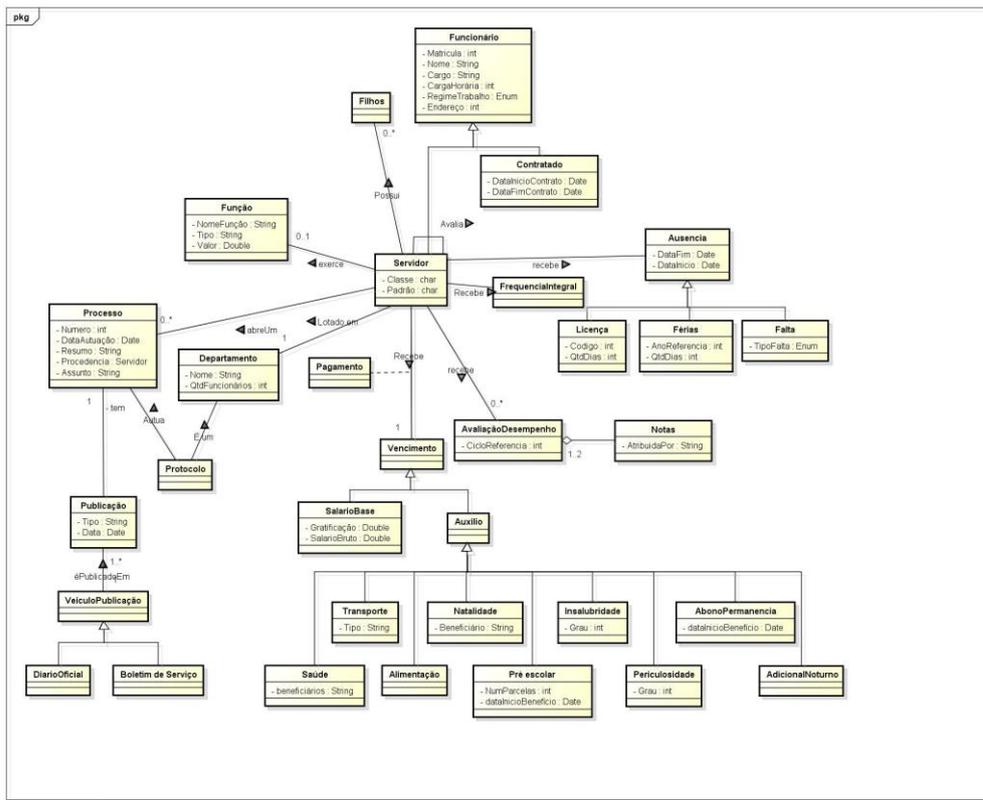


Figura 6: Modelagem do Cenário de testes

4.3 Criação do cenário sem utilização de Padrões de Projeto

Neste caso, foi utilizado o Software Protégé para se criar a ontologia do cenário a ser estudado. Uma vez que nesta etapa não haverá reutilização de Padrões pré-existentes, a ontologia deverá ser criada desde o início com suas particularidades, como por exemplo, as suas propriedades e relações. Uma boa prática é seguir as regras mencionadas na Seção 3.4 desde trabalho para criação do Modelo.

De acordo com essas boas práticas a primeira questão a ser respondida são as perguntas de “Quem faz o quê, quando e onde?”. Pode-se definir a entidade Servidor e seus relacionamentos e tarefas, uma vez que essa entidade é o ponto de partida. Já para tratar “Quais são as partes de uma coisa” foram criadas as entidades que iram conceituar as tarefas que se deseja representar, como por exemplo, as classes menos específicas de conceitos mais gerais, até mesmo as classes mais específicas como os Tipos de Auxílio ou os Tipos de Ausência. No caso da entidade que trata a frequência do Servidor, verifica-se “Qual é o período de tempo”. Ou seja, a cada pergunta dos casos genéricos respondidas, uma parte do cenário de testes foi sendo moldada até que se chegasse à criação do todo.

A Figura 6 ilustrou a criação deste modelo, ou seja, das entidades que compõe o Cenário de Testes. Neste modelo, por não haver uma base inicial para se criar o modelo, a ontologia é criada sem nenhum padrão. Como mencionado anteriormente, é necessária a criação das propriedades de cada entidade também os relacionamentos da ontologia, além das instancias que se deseja criar, podendo definir as propriedades do objeto, ou seja, as relações entre as entidades, e as propriedades do arquivo, que são os atributos destas entidades, como mostra na Figura 7. Também é possível definir anotações, além de outras propriedades, como o Domínio ou classe de equivalência (Descrição).

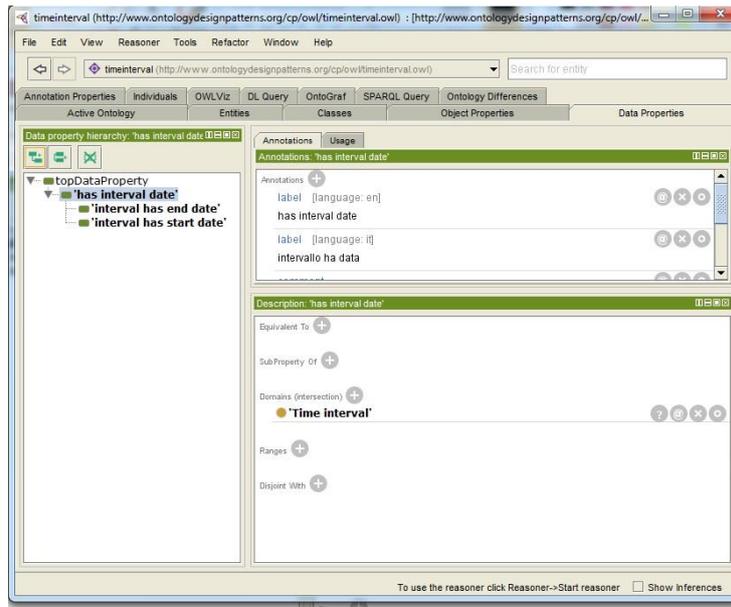


Figura 7: Janela *DataProperty* do Padrão *TimeInterval* no Protégé

4.4 Análise da modelagem sem uso de padrões

Conclui-se que, ao se criar este modelo sem a utilização de padrões, embora a ideia inicial fosse de simplicidade e expressividade por se criar um modelo independente de padrões pode-se notar justamente o oposto. Esta modelagem do cenário sem utilização de padrões de projeto de ontologia foi um primeiro teste de criação desta ontologia, e, portanto foi possível ver a dificuldade na definição de conceitos desenvolvidos, e na escolha das propriedades de cada um desses conceitos. Também foi possível verificar que ao se criar um modelo desde o início, sem usar padrões, a dificuldade para fazer a verificação em relação à adequação do modelo ao cenário proposto (foi modelado da melhor forma possível?), ou seja, a complexidade em se trabalhar com conceitos e propriedades aumenta.

Em relação às linguagens pesquisadas para utilização neste trabalho, foi visto, por exemplo, que relações genéricas (sem domínio e imagem pré-definidos) em modelos OWL não podem ser representadas em UML, da mesma forma, algumas propriedades dos relacionamentos (equivalência) também só estão explicitamente definidas na linguagem OWL. Apesar disto, para a modelagem do cenário deste trabalho essas relações não precisaram ser modeladas, e com isso a escolha de se utilizar a linguagem UML se mostrou suficiente para representar os conceitos desejados.

Após a modelagem do cenário sem o conhecimento dos padrões já existentes é possível notar a dificuldade de se perceber a expressividade dos conceitos que devem ser modelados, e a extensão dos mesmos (geraram uma informação nova, ou acrescentaram dados à informação que está sendo descrita?). Embora os conceitos pareçam "óbvios" ao se conhecer o "minimundo", a grande expressividade permitida pelo uso de ontologias mostra o quanto é difícil dominá-los, ou seja, é como se cada conceito na verdade desencadeasse vários outros conceitos por trás dele que podem ser retratados durante esta modelagem e é muito difícil saber até qual ponto cada um destes conceitos devem ser "aprofundados" para poder passar a ideia que se deseja.

Outra dificuldade encontrada está na percepção do quanto um conceito é independente do outro, ou seja, em qual ponto um conceito que se deseja modelar é na verdade somente uma propriedade ou atributo daquele conceito ou se na verdade ele é mais do que isso, ele é outro conceito que deva ser expandido para retratar uma tarefa deste modelo. Ao se observar os padrões existentes, muitas vezes é possível identificar essas respostas, pois elas já foram modeladas nos padrões já existentes.

Em resumo, a modelagem sem os padrões, mas seguindo as melhores práticas sugeridas no Capítulo 3 do desenvolvimento de ontologias, se torna muito mais complexa por não dispor de nenhuma estrutura base.

4.5 Criação do cenário usando Padrões de Ontologias

Nesta etapa do projeto serão propostas algumas instanciações ao cenário proposto inicialmente utilizando os padrões já descritos no Capítulo 3 deste documento. Os padrões considerados estão disponíveis para consulta e utilização no repositório de padrões de projeto de ontologias escolhido como base para este trabalho, no site <http://ontologydesignpatterns.org/>.

Nesta seção serão ilustrados os padrões, mostrando, quando disponíveis, o diagrama padrão, retirado do site do repositório, a modelagem do conceito e como ele ficaria após a adaptação do padrão.

Após a análise dos padrões disponíveis no repositório para instanciação, o primeiro a ser identificado para uso no cenário modelado foi o padrão *Person*, desenvolvido por

Aldo Gangemi, como é ilustrado na Figura 8, uma vez que a entidade Funcionário é utilizada para retratar pessoas físicas, e este grupo pode ser decomposto por duas subclasses de membros distintos, os funcionários servidores, e os funcionários contratados.

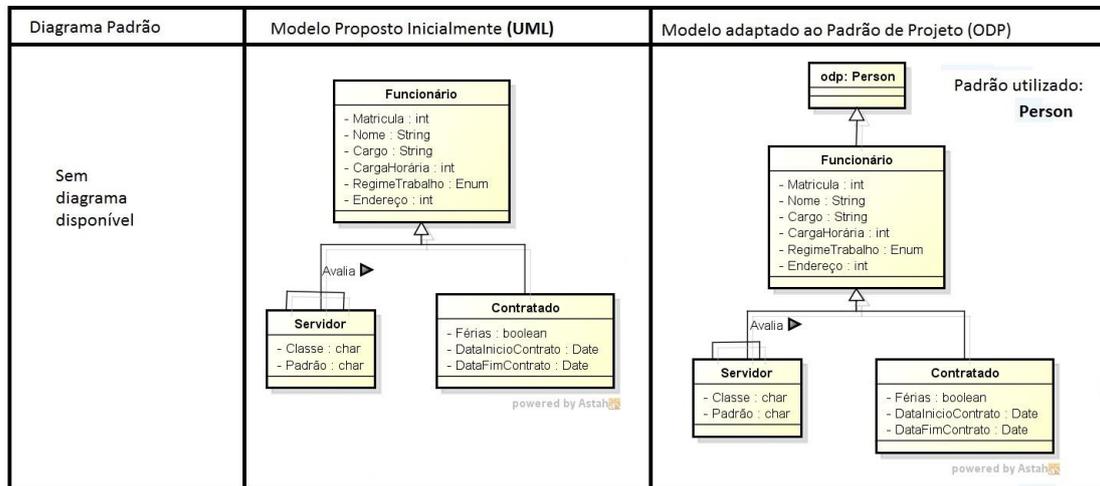


Figura 8: Padrões para as classes Funcionário - Servidor – Contratado

A Figura 9 mostra o uso do padrão *Person* para definir que o funcionário é uma pessoa. Uma pessoa, além de ser um indivíduo (pessoal Natural) também é um funcionário (pessoa Social), ou seja, exerce um papel social. Este papel que ela exerce em uma empresa é o seu cargo e este pode ser conceituado, por ter características próprias.

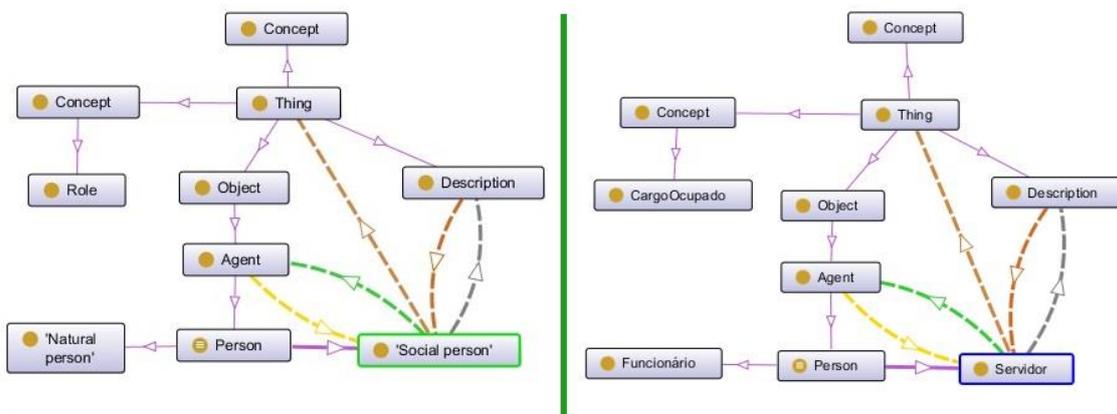


Figura 9: Comparativo entre o padrão Person e seu uso visto no Protegé

Ainda tratando da classe Servidor, podemos retratar a forma que um servidor pode exercer uma função, como por exemplo a função de chefia. Neste caso usamos o padrão *ParticipantRole*, desenvolvido por Eva Blomqvist, para esta representação, como ilustra a Figura 10.

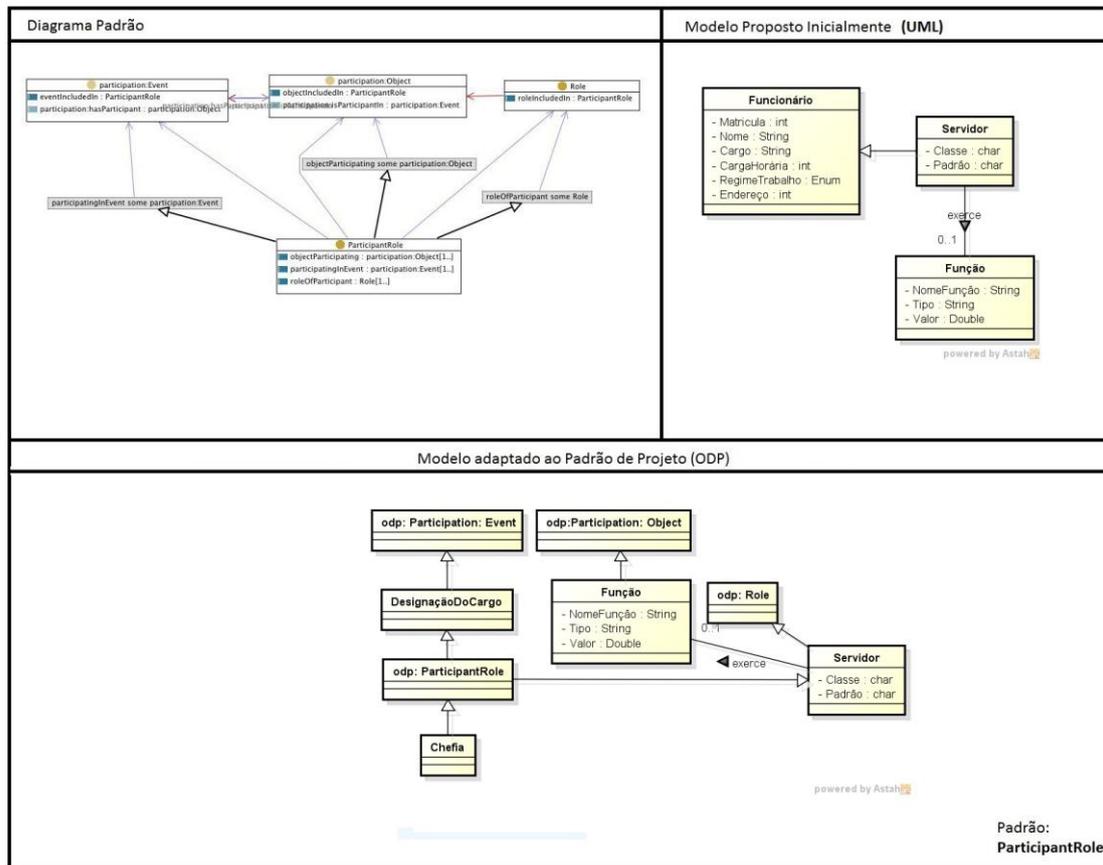


Figura 10: Padrões para as classes Função – Servidor

A Figura 11 mostra o uso do padrão *ParticipationRole* para definir que o funcionário Servidor exerce um papel em um determinado evento. Esse papel, no caso do cargo de chefia, é a função que um servidor irá exercer desde sua nomeação até enquanto não for nomeado outro funcionário designado para o cargo.

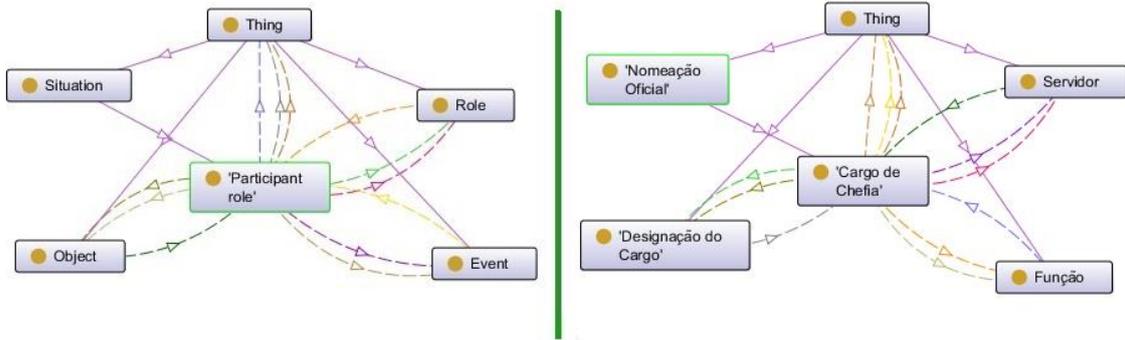


Figura 11: Comparativo entre o padrão *ParticipantRole* e seu uso visto no Protegé

Para a adaptação das classes que representam qual a frequência mensal um servidor pode receber, ou seja, se ele compareceu ao seu departamento todos os dias, ou se faltou, ou se o mesmo encontrava-se em férias, ou em período de licença médica a melhor alternativa seria a utilização do padrão de intervalo de tempo (*TimeInterval*), proposto por Valentina Presutti, uma vez que pode-se criar uma representação para os 3 tipos de ausência um servidor pode possuir em um determinado período, que são os conceitos de férias, de licença e de falta, além da possibilidade do servidor ter sua frequência integral, como pode ser visto na Figura 12.

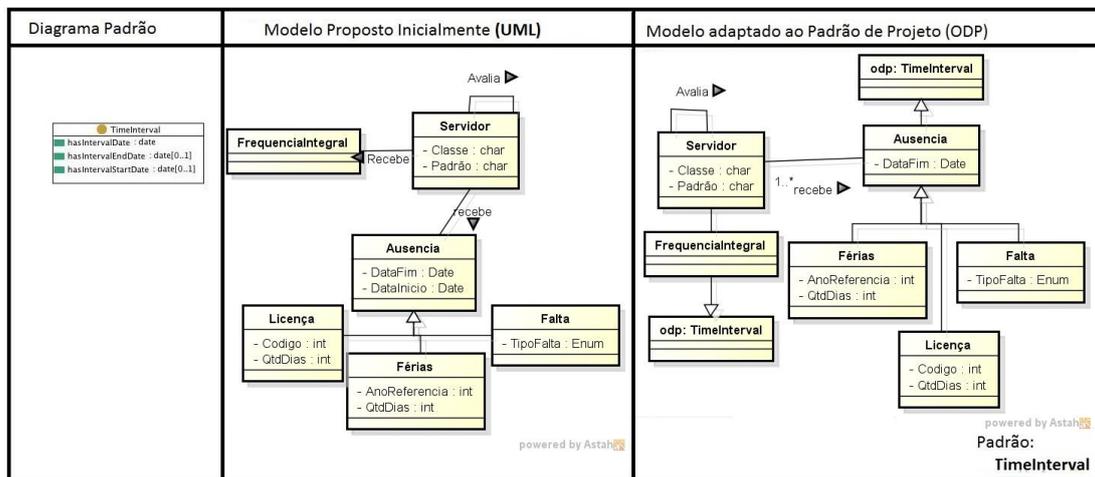


Figura 12: Padrões para representar as ausências de um servidor

A Figura 13 mostra essa aplicação do padrão *TimeInterval*.

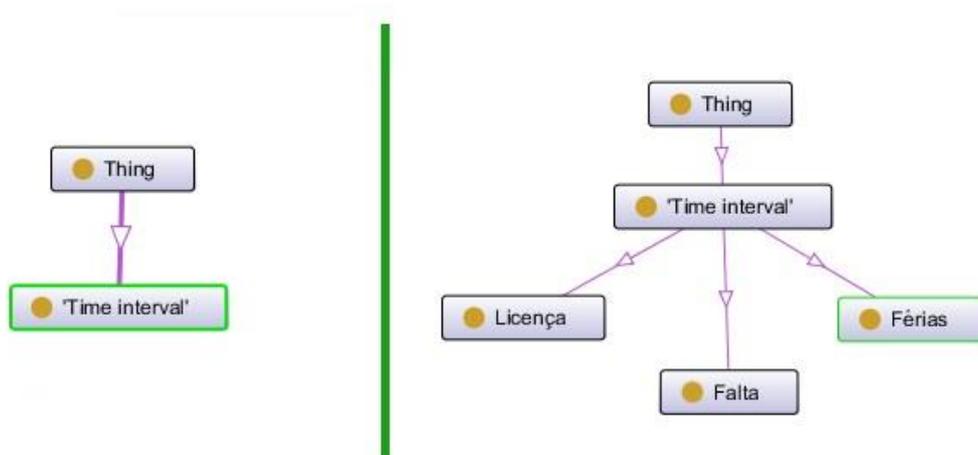


Figura 13: Comparativo entre o padrão *TimeInterval* e seu uso no Protégé

Para representar o pagamento recebido por um servidor, seja do seu salário base, ou de algum auxílio que o mesmo esteja recebendo, o padrão *Participation* para mostrar como o evento Pagamento gera o objeto vencimento.

Este padrão ilustra a participação de um objeto em um evento como mostram as Figura 14 e 15.

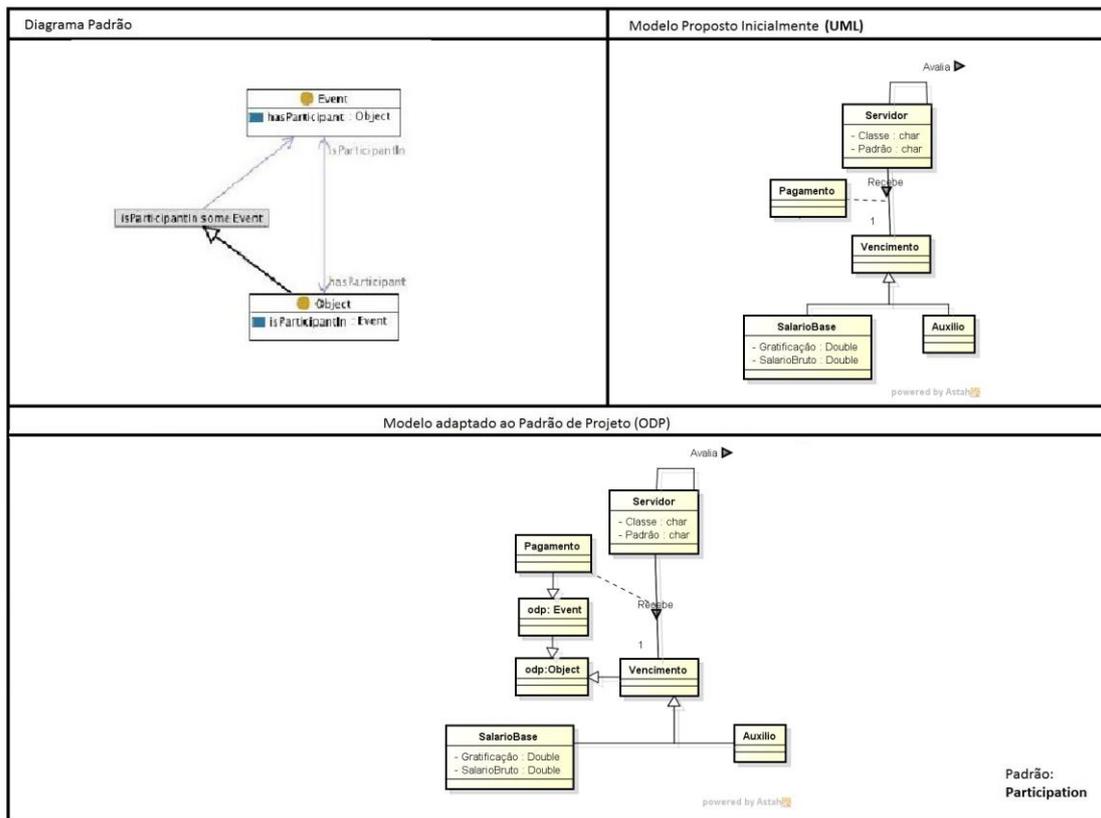


Figura 14: Padrão para representar o Pagamento ou Auxilio recebido pelo servidor

A classe processo também foi modelada usando este padrão *Participant*, e, portanto só foi representada no cenário final.

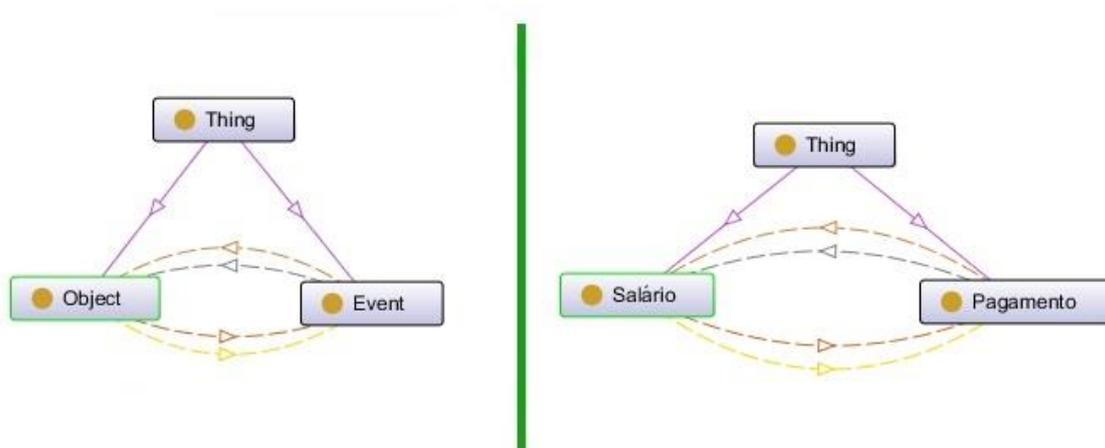


Figura 15: Comparativo entre o padrão *Participation* e seu uso no Protégé

Todo servidor, em determinada época (neste exemplo o evento abordado foi a Avaliação

de Desempenho) recebe um conjunto de diferentes notas, ou seja, o padrão *Event* é usado para representar o evento “Avaliação de Desempenho” do qual uma coleção de notas (padrões *Collection* e *Set*) é atribuída a um servidor. A Figura 16 mostra como adequar estes padrões *Event*, *Collection* e *Set*, desenvolvidos por Eva Blomqvist neste contexto.

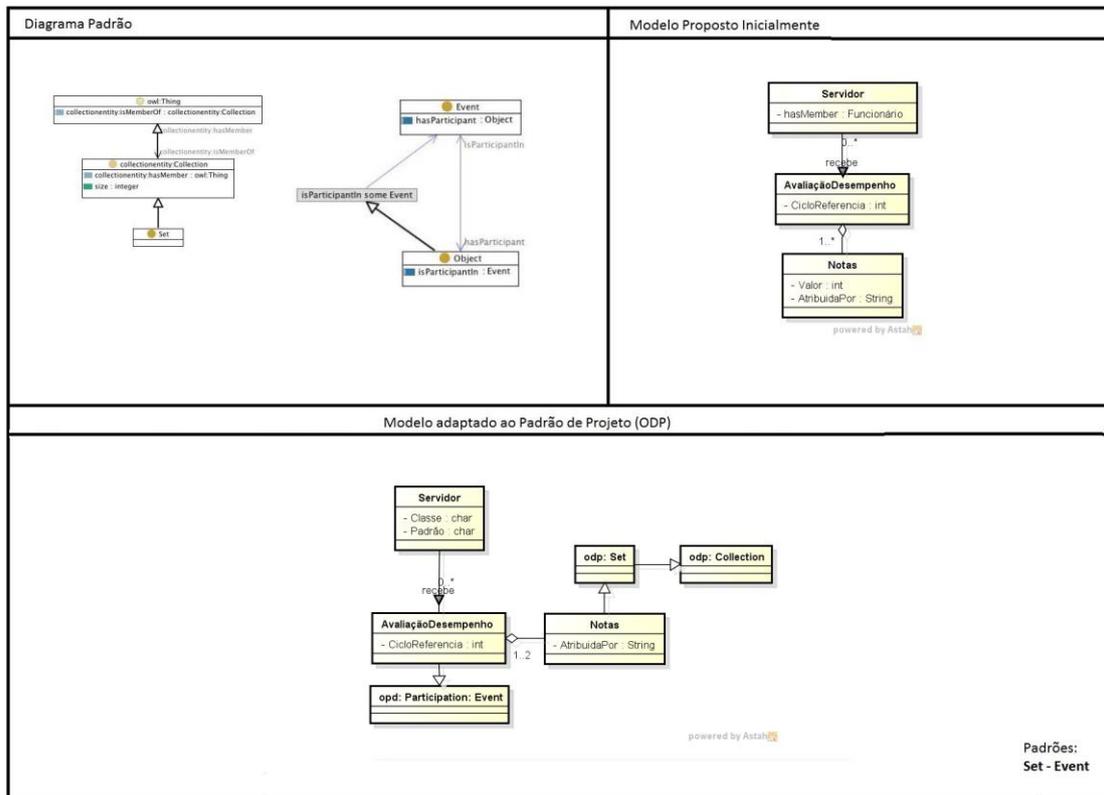


Figura 16: Padrão para as classes Servidor – Notas

Todas as atividades ocorridas dentro de um Departamento Pessoal devem ser registradas em um veículo de comunicação, e portanto cada uma destas atividades gera uma publicação. Existem dois meios de comunicação usados para este fim, o Boletim de Serviço e o Diário Oficial. Para representar a publicação deste processo será usado Padrão *Place*, desenvolvido por Aldo Gangemi, como mostra a Figura 17.

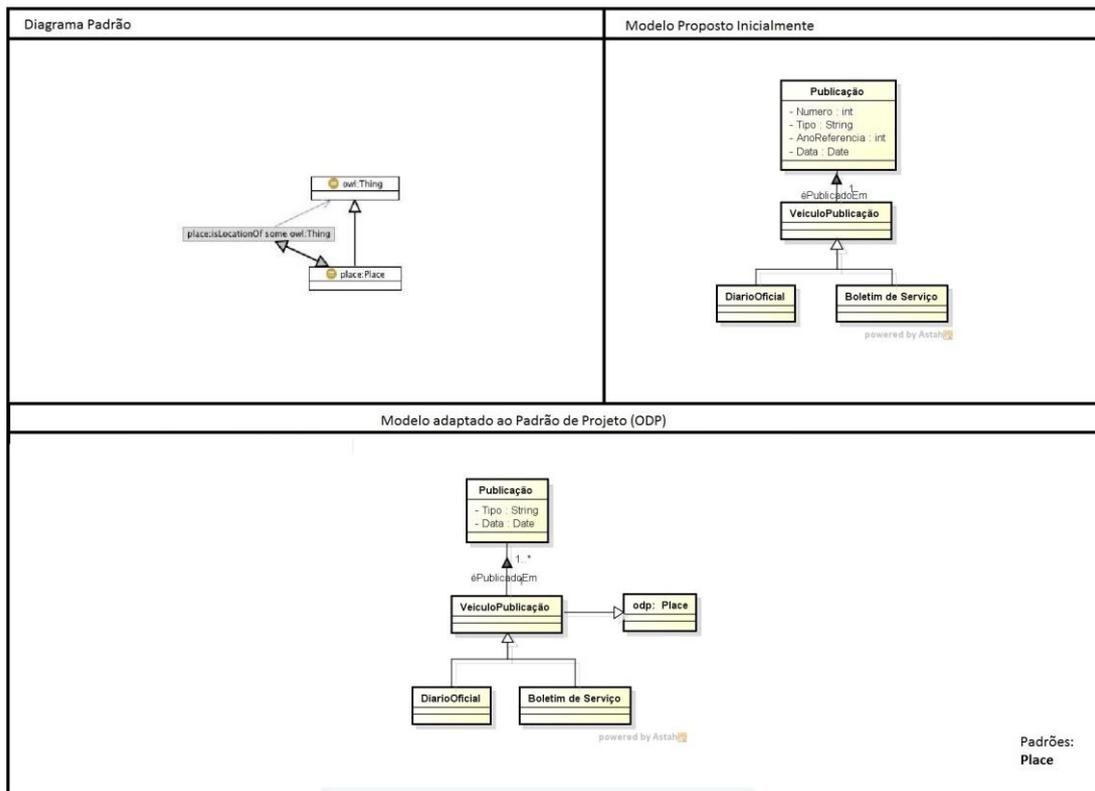


Figura 17: Padrão *Place* para definir os locais de Publicações

Por fim, a Figura 18 ilustra o resultado final, com as devidas adaptações, após a utilização dos padrões de projeto de ontologia.

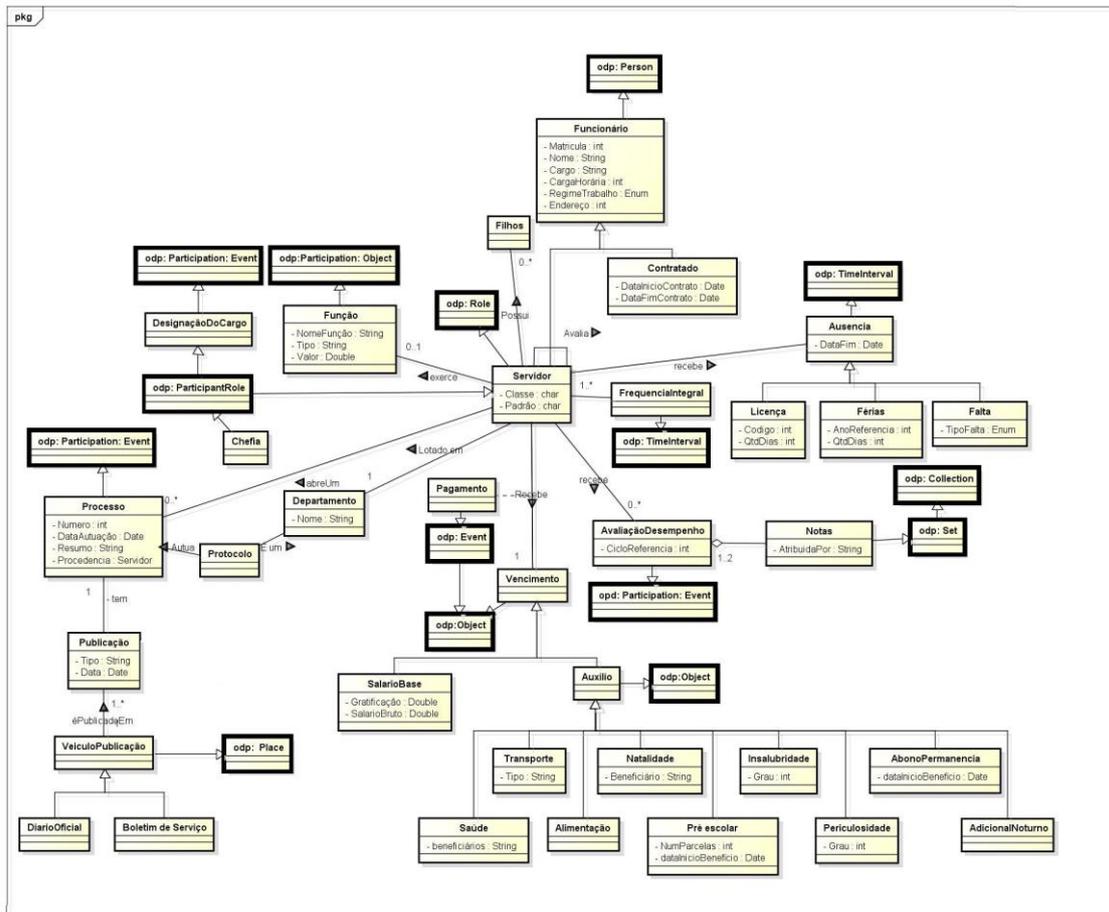


Figura 18: Cenário Proposto para o uso de Padrões

4.6 Análise sobre o uso de padrões de Ontologias

Após recriar este cenário utilizando os Padrões é possível perceber os benefícios de se utilizá-los. A introdução de novas classes aumentou a quantidade de informações disponíveis, aumentando assim a semântica agregada ao modelo, o que pode ser visto na modelagem da função exercida pelo servidor, na qual foram acrescentadas as informações referentes à designação de cargo e a explicitação da classe chefia. Os padrões também esclarecem os papéis desempenhados por cada classe, como no caso do servidor desempenhar um papel (role). Ao ocorrer o evento pagamento, o uso de padrões também explicitou a ideia de que existe um vencimento e este é composto por dois objetos: O salário base e os auxílios. Não só pela facilidade em entender a estrutura base necessária ao projeto, é simplificado o processo de ambientação do usuário ao cenário que será modelado e assim pode-se elucidar possíveis dúvidas em relação à expressividade de determinados conceitos que se deseja abordar.

Outro fator que pode ser observado na modelagem com os padrões de projeto está na possibilidade de se recorrer às regras das boas práticas para utilização de ontologias já mencionadas para identificação de qual padrão melhor se adequa a cada situação durante a modelagem e adaptação da ontologia, por exemplo, para escolher o padrão de ontologia mais indicado, exemplo padrão “*Place*” para responder “Qual é o lugar de alguma coisa?”.

Durante a realização deste trabalho, foi possível notar que ao se importar os padrões (disponíveis no repositório escolhido e já na modelados na linguagem OWL) com o auxílio de uma ferramenta, o *Protégé*, a análise das classes, das relações e dos atributos já modelados pode fazer uso dos comentários disponíveis em cada padrão, para facilitar a escolha e esclarecer dúvidas quanto ao uso dos padrões,

A criação do modelo inicial através da linguagem UML, mais conhecida e utilizada para modelagem, também foi um fator que facilitou a abordagem deste projeto. Uma vez que os padrões já são disponibilizados em OWL no repositório online utilizado, foi mais fácil reutilizar os mesmos ao adaptá-los ao modelo criado.

A grande dificuldade neste projeto esteve em entender a representação dos padrões de Ontologia na linguagem OWL. No momento existe algum conteúdo teórico sobre este assunto, mas nenhum conteúdo prático foi encontrado para ser utilizado ao iniciar esta abordagem. Durante a fase de pesquisas deste projeto foi possível notar a dificuldade em se obter manuais ou trabalhos anteriores para criação e reutilização de padrões, ou mesmo como se devem representar estes padrões, por exemplo, trabalhos anteriores de adaptação dos ODPs a cenários pré-existentes. Segundo o Trabalho de Eva Blomqvist [2009], para se utilizar os conceitos dos padrões em um modelo devem ser utilizadas subclasses, porém o assunto não é aprofundado, ou seja, nenhum trabalho faz referência de como se deve representar graficamente o uso de padrões de ontologia para a modelagem de um cenário específico.

Em relação à utilização de repositórios, a dificuldade esteve em se importar os padrões diretamente para o Software escolhido. Nenhuma ferramenta que fizesse essa ligação entre o repositório e o software de criação de ontologias foi encontrada.

Outra dificuldade foi escolher qual o melhor padrão, uma vez que um conceito pode ser

representado por mais de um padrão, pois não existe uma regra para essa escolha.

5 Conclusão e Trabalhos Futuros

Neste trabalho foi possível verificar quais as vantagens obtidas ao se modelar um sistema utilizando padrões de projeto de ontologia, disponíveis para reutilização em repositórios próprios para isso. Foi utilizado o site *www.ontologydesignpatterns.org* para buscar os padrões criados pela comunidade dos desenvolvedores de ontologias, e baseado em suas próprias experiências anteriores, que melhor se encaixariam ao cenário estudado.

A ideia deste trabalho foi criar uma documentação contendo informações sobre o cenário estudado através da modelagem conceitual dos dados que são relevantes em um Departamento Pessoal. Essa modelagem foi realizada de acordo com as melhores práticas de desenvolvimento de uma ontologia

O primeiro passo escolhido para esse trabalho foi criar um modelo UML do cenário. Ao se criar esse modelo foi possível ter uma noção do que seria necessário tratar inicialmente (qual seria a abrangência do modelo), ou seja, criar uma estrutura-base. Ao iniciar a modelagem através dos padrões de projeto de ontologia, problemas foram identificados e tratados para que a adaptação pudesse ser possível da seguinte forma: Verificou-se a necessidade de criação de novas classes para se abordar as tarefas desenvolvidas, ou mais bem defini-las, além da adaptação de outras para que seja possível representar melhor as diversas situações envolvidas no cenário, como por exemplo, para modelar a classe inicialmente chamada de função. Na modelagem sem a utilização dos padrões, percebeu-se a necessidade de criar o conceito geral (um servidor possui uma função). Essa ideia, para o leitor que não conhece o cenário modelado não explica muita coisa, mas, ao se modelar usando os padrões foi possível acrescentar mais informações como, por exemplo, que essa função é um papel que o servidor exerce a partir de um determinado evento.

Assim, é possível perceber que a utilização de padrões de projetos de ontologias facilitou a garantia de uma maior qualidade e expressividade da ontologia desejada. Também foi possível mostrar como é possível prevenir erros estruturais e confusões iniciais da ideia que se deseja passar com o modelo criado. Ao se utilizar um padrão genérico você pode se familiarizar melhor com o cenário que será tratado e com quais

as melhores soluções para o modelo desejado. Ainda que os modelos sejam genéricos eles permitem montar uma boa estrutura para o projeto, uma vez que os conceitos de cada padrão estão mais bem definidos.

Quando o objetivo é a criação de um modelo o mais fiel possível ao cenário desejado, uma combinação entre a utilização de padrões e a adaptação de modelos de padrões é a melhor solução. É possível usar os padrões para modelar a ideia inicial do projeto, ou seja, a estrutura do mesmo. Caso não exista um padrão que se adapte a determinada ontologia e que deva ser representada no projeto ela pode ser adaptada posteriormente, como visto por Blomqvist [2009] que menciona que os padrões podem ser utilizados como modelos, ou soluções parciais para propor o início de uma solução para o problema em questão. Uma vez que a base do modelo esta pronta, e assim o usuário do projeto já está mais familiarizado com a estrutura, será mais fácil aperfeiçoá-la e/ou incrementá-la.

E, se por um lado, essas são as vantagens de quando se utiliza os padrões disponíveis em um repositório, por outro podemos mencionar as vantagens de se "abastecer" um repositório.

Ao se disponibilizar padrões, pode se contribuir com suas próprias experiências de projeto, tornando o repositório mais rico em informações a disposição de usuários.

Um exemplo disto são os repositórios de domínios específicos, que contam com diversas ontologias disponíveis para acesso online, o que garante “um acervo maior de ontologias para consulta” no momento da criação de um modelo deste domínio específico.

Inclusive, é importante mencionar que se verificou essa dificuldade neste trabalho, uma vez que o repositório utilizado contava com uma média de 100 padrões de ontologia disponíveis para uso, ou seja, muitos dos conceitos necessários ao modelo não estavam acessíveis no conjunto de padrões existentes no repositório.

O repositório utilizado neste projeto conta, atualmente, com diferentes colaboradores cadastrados que podem submeter padrões, assim como sugerir alterações, além de

participar como uma comunidade voltada a assuntos referentes a padrões de ontologias no site do repositório online.

Este trabalho deixa como proposta para um futuro trabalho a implementação do cenário utilizado em uma ferramenta própria a linguagem OWL com todos os seus padrões de maneira otimizada como mostra o trabalho Lantow *et al.* [2013].

Outra proposta deixada por este trabalho é um comparativo entre os repositórios de padrões de ontologias existentes, analisando suas vantagens e desvantagens, por exemplo, referentes à interface de usuário, domínios abrangidos pelo repositório, e participação dos usuários.

Há também a ideia de se trabalhar com uma modelagem modularizada, ou seja, aumentando/diminuindo o nível de abstração dos conceitos envolvidos no modelo, como por exemplo, para representar o pagamento de um auxílio como uma atividade desencadeada pela abertura de um processo. Assim, detalhando a ontologia criada para aumentar a expressividade a cada camada, a abrangência do domínio seria maior e mais precisa, de acordo com o domínio individual de cada tarefa abordada, analisando seus requisitos, como sugerido no trabalho de Lantow *et al.* [2013].

Referências Bibliográficas

- Aranguren, M. **Ontology Design Patterns for the Formalisation of Biological Ontologies**, Report for the degree of Master of Philosophy in the Faculty of Engineering and Physical Sciences, 2005.
- Azevedo, L.; Lopes, M.; Souza, J.; Siqueira, S.; Cappelli, C. ; Baião, F. 2008 **Uma Metodologia de Avaliação de Ferramentas para Gestão de Ontologias** Disponível em <http://www.uff.br/ontologia/artigos/210.pdf> .
- Azevedo, L.; Lopes, M.; Souza, J.; Siqueira, S.; Cappelli, C. ; Baião, F. ; Santoro, F.; Nunes, V. ; Magdaleno, A. 2008 **Gestão de Ontologias**. Disponível em https://www.academia.edu/2891136/0002_2008-Gestao_de_Ontologias
- Ceci, F.; **Um Modelo Semiautomático Para A Construção E Manutenção De Ontologias A Partir De Bases De Documentos Não Estruturados, 2010**
- Egaña, M.; Rector, A.; R Stevens, R.; Antezana, E. **Applying Ontology Design Patterns in Bio-ontologies**, 2008.
- Blomqvist, Eva. **Semi-automatic ontology construction based on patterns**. 2009.
- Gangemi, A.; Presutti, V. **Ontology Design Patterns**. Handbook on Ontologies (Second Edition). 2004.
- Gangemi, A.; Presutti, V.; **Pattern-based ontology design**, Apresentação no Semantic Technology Lab, 2009
- Gangemi, A. **Ontology Design Patterns for Semantic Web Content**, In The Semantic Web–ISWC 2005 (pp. 262-276). Springer Berlin Heidelberg. 2005.
- Gangemi, A; Gomez-Perez, A.; Presutti, V. e M.C Suarez-Figueroa. **Towards a Catalog of OWL-based Ontology Design Patterns**. CAEPIA 2007
- Grigoris A.; Harmelen F.; **Web Ontology Language: OWL**. Handbook on Ontologies (Second Edition). 2004.
- Gruber, T.; **A Translation Approach to Portable Ontology Specifications**, Technical Report, 1993.
- Grüninger, M.; Hahmann, T.; Hashemi, A.; Ong, D. **Ontology Verification with Repositories**, 2010.
- Grüninger, M.; Hahmann, T.; Hashemi, A.; Ong, D. e Ozgovde, A. **Modular first-order ontologies via repositories**, Applied Ontology, 7:169-210, 2012

Guarino, N.; Oberle, D.; Steffen Staab, S. **What Is an Ontology?** Handbook on Ontologies (Second Edition). 2004.

Guedes, A. Um **Framework de Padrões de Projeto**, 2013.

Hashemi, A.; Gruninger, M. **Ontology Design Through Modular Repositories**, 2009.

Morais, E, Ambrósio, A.; **Ontologias: conceitos, usos, tipos, metodologias, ferramentas e linguagens**, Technical Report, 2007.

Mortensen, J.; Horridge, M.; Musen, M.; Noy, N. **Modest Use of Ontology Design Patterns in a Repository of Biomedical Ontologies**, 2012.

Presutti, et al.: **A Library of Ontology Design Patterns: Reusable Solutions for Collaborative Design of Networked Ontologies**. NeOn D2.5.1. 2008.

Bhogal, Jagdev; Macfarlane, Andy; Smith, Peter. **A review of ontology based query expansion**. Information processing & management, v. 43, n. 4, p. 866-886, 2007. Apud Santos, V. **Uma Arquitetura Suportada Por Busca Semântica Para Recuperação De Fontes De Informação Em Repositórios De Metadados**, 2011. Santos, V. **Uma Arquitetura Suportada Por Busca Semântica Para Recuperação De Fontes De Informação Em Repositórios De Metadados**, 2011

Staab ,S.; Studer ,R. **Handbook on Ontologies**, Second Edition. 2004.

Fensel, Dieter et al. **Ontologies and electronic commerce**. Intelligent Systems, IEEE, v. 16, n. 1, p. 8-14, 200 apud Sell, Denilson et al. **Uma arquitetura para business intelligence baseada em tecnologias semânticas para suporte a aplicações analíticas**. 2006.

Gomez-Perez, Asuncion; Fernández-López, Mariano; Corcho, Oscar. Ontological engineering. Heidelberg: Springer, 2004. Apud Sell, Denilson et al. **Uma arquitetura para business intelligence baseada em tecnologias semânticas para suporte a aplicações analíticas**. 2006. Sell, D.; **Uma Arquitetura Para Business Intelligence Baseada Em Tecnologias Semânticas Para Suporte A Aplicações Analíticas**, 2006. Tese (doutorado)

Gangemi, A.; Gómez-Pérez, A.; Presutti, V.; Suárez-Figueroa, M.; **Towards a catalogo of OWL-Based Ontology Design Patterns**. 2007. Disponível em http://oa.upm.es/5212/1/Towards_a_Catalog_of_OWL-based_Ontology_Design_Patterns.pdf

Daltio, J.; Medeiros, C. **Aondê: Um Serviço Web de Ontologias para Interoperabilidade em Sistemas de Biodiversidade**. 2008. Anais do XXVIII Congresso da SBC.

Mendonça, F.; Cardoso, A.; Drumond, E.; **Ontologia de aplicação no domínio de mortalidade: uma ferramenta de apoio para o preenchimento da declaração de óbitos.** 2010. *Ciência da Informação*, 2011, vol. 39, n. 3, pp. 23-34. [Artigo de Revista]

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.; **Design Patterns: Elements of Reusable Object-Oriented Software**, 1997.

Martins, M.; Librelotto, G.; **Comparando As Linguagens Para Representação De Ontologies.** 2007. *Em Disc. Scientia. Série: Ciências Naturais e Tecnológicas*, S. Maria, v. 8, n. 1, p. 119-130, 2007

Lantow, Birger; Sandkuhl, Kurt; Tarasov, Vladimir. **Selecting Content Ontology Design Patterns for Ontology Quality Improvement.** *on Logi ogies in*, p. 68, 2013.

Vieira, R. et al. **Web semântica: ontologias, lógica de descrição e inferência.** *Web e Multimídia: desafios e soluções.* Porto Alegre: SBC, p. 127-167, 2005.

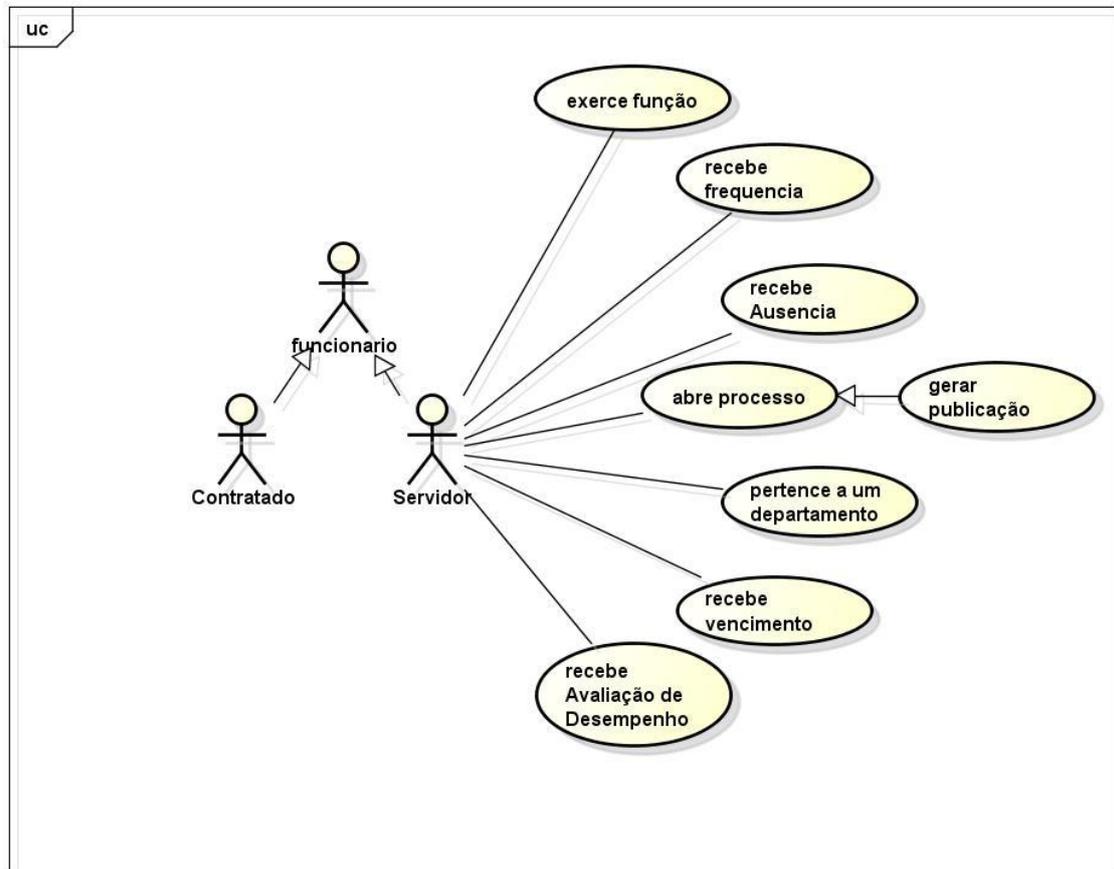
Engiel, P; Araújo, R; Cappelli, C. 0016/2011-**Apresentação de modelos de processos públicos—uma discussão sobre abordagens de representação de processos.** *RelaTe-DIA*, v. 5, n. 1, 2011.

Cranefield, S; Purvis, M. **UML as an ontology modelling language.** 1999.

Moody, D.; Shanks, G. **Improving the quality of data models: empirical validation of a quality management framework.***Information systems*, v. 28, n. 6, p. 619-650, 2003.

Castro, L.; Baião, F; Guizzardi, G. **Abordagem Linguística para a Modelagem Conceitual de Dados com Foco Semântico.** 2010. Tese de Doutorado. MSc Dissertation, Unirio, Rio de Janeiro.

Anexo – Diagrama de casos de uso



powered by Astah