

ECI046 – Ontologias em Organizações
Prof.: Renato Fabiano Matheus
Supervisão: Maurício Barcellos

Atividade Avaliativa 02 - Exercício individual

Versão 20181025 (as modificações feitas após versão inicial estão **marcadas)**

Prazo de entrega: 05/11/2018 até 23h55 Valor: 40 pontos Entrega via Moodle.

Obs: entrega com atraso implica subtração de 3 pontos a cada dia. Entrega em 10/11/2018 até 23h55 valendo 3 pontos.

Descrição da atividade

1. Especificar um **problema organizacional** a ser resolvido utilizando **ontologias computacionais**, conforme detalhado no item **Especificação** deste documento.

Passos preliminares

2. Criar uma cópia deste documento no Google Drive e editar a cópia como seu documento de entrega
 - URL do documento:
(URL do [documento base](#))
(Usar menu File ⇒ Make a Copy do Google Drive com usuário Google conectado e depois fazer SHARE ⇒ “Get Shareable Link” ⇒ “Done” e depois copiar endereço do documento a seguir)
<Endereço deste documento no Google Drive>
 - **ENTREGA: documento com respostas em formato PDF via Moodle.**
 - Sugere-se colocar também uma cópia do PDF no seu Github: <endereço github>
3. Identifique-se: Aluno: Luiz Antônio Lourenço da Silva

Requisitos de arquitetura da ontologia (a implementação das ontologias será objetos das Atividades 03 e 04)

4. Criar uma nova ontologia OWL básica em RDF/XML usando Protégé e/ou [Protégé Web](#), cujo nome deve estar relacionado com a organização e o problema cuja solução você irá modelar e implementar. **Os nomes das classes e propriedades de sua ontologia base devem ser em português**
5. Agregar à sua ontologia básica pelo menos outras 2 (duas) ontologias vistas durante o curso ou disponíveis na Web, e.g.: Schema.org, FOAF, DBpedia Linked Data, SKOS, BFO e OBO-Foundry, ... (ver [slides](#) usados em aulas).
6. Sua ontologia base deve conter pelo menos 5 classes, cada classe pelo menos 3 atributos e 3 consultas

SPARQL. As consultas SPARQL devem consultar preferencialmente pelo menos 2 classes.

7. Lembre-se de usar restrições de propriedades OWL (InverseOf, SameAs, DistinctWith, Min/Max) (ver apresentações sobre OWL).
8. Procure usar outras características para propriedades de dados (“lang”, com diferentes línguas “en”, “pt”; tipos de dados “string”, “integer”, outros).
9. Não utilizar como base a ontologia universidade.owl.

Especificação básica

Especifique (cada especificação a seguir deve ser feita em 1 ou 2 parágrafos, com 10 a 20 linhas):

10. **(6 pts) Cenário** (descreva o contexto e a organização na qual o problema organizacional será resolvido) (e.g., biblioteca, agência bancária, loja de roupas presencial ou virtual)
11. **(6 pts) Processo de trabalho** (identifique e descreva o processo de trabalho que será foco da solução proposta) (e.g., processo de controle de usuários, processo de controle de estoque, processo de venda, ...)
12. **(6 pts) Problema a ser resolvido** (descreva o problema) (e.g., “Controlar quais usuários estão com livros emprestados”; “Identificar quais usuários estão com entregas em atraso”)

As especificações a seguir devem ser apresentadas em tabelas com vários itens cada:

13. **(3 pts) Requisitos de software** a serem implementados e **forma de implementação** (mínimo de 3 requisitos específicos para “**problema a ser resolvido**”) (criar tabela) (e.g., Especificar formato de dados ⇔ Criação de ontologia; **Identificar/Listar usuários/produtos que são do tipo X/que custam mais do que Y...** (procure ser específico neste requisito no sentido de ser capaz de fazer uma consulta SPARQL na sua ontologia para resolvê-lo (nas tabelas seguintes e na Atividade Avaliativa 03), sendo que a indicação é que as consultas SPARQL acessem mais de uma classe da sua ontologia e das ontologias agregadas ⇔ Cadastro de indivíduos usando Protégé OWL ou Protégé Web; Consultar usuários em atraso ⇔ “Fazer consulta SPARQL”)
14. **(3 pts) Modelagem de dados** (identificar em quais ontologias/classes/propriedades cada um dos requisitos irão impactar) (criar tabela à parte ou incorporar tabela de **Requisitos de software**)
15. **(3 pts) Perfil de usuários** (criar tabela de funcionalidades por usuário) (identificar perfis de usuários do sistema e as várias funcionalidades que cada um poderá usar) (e.g., Administrador ⇔ Criação ontologia OWL com Protégé, Gerente, Estagiário ⇔ Consultar, ...)
16. **(3 pts) Requisitos de interface** (identificar como será a interface para acesso às funcionalidades) (criar tabela) (associar Requisitos de software ⇔ Usuário(s) ⇔ Descrição de requisitos de interface com identificação de ambiente) **(identificar parâmetros de entrada e saída)**

Solução (coloque suas respostas a partir daqui)

Para facilitar a modelagem nas Atividades seguintes, procure **marcar aquelas palavras que se tornarão classes, propriedades ou restrições** de consultas SPARQL em negrito no texto de sua solução. Por exemplo, “**Cenário**: as Pessoas relacionam-se dentro da organização por meio de um aplicativo de **Mensagens...**” ou “**Cenário**: o **parque de diversões** é um negócio no qual relacionam-se **Funcionários, Clientes e Fornecedores...**”. Essa estratégia vai facilitar a identificação de quais palavras correspondem a classes da ontologia.

Especifique (cada especificação a seguir deve ser feita em **1 ou 2 parágrafos, com 8 a 20 linhas**):

EXEMPLO DELINEAMENTO PARCIALMENTE RESPOSTAS POSSÍVEIS (a seguir)

Especificação básica

Cenario

17. **Cenário** (descreva o **contexto** e a **organização** na qual o problema organizacional será resolvido) (e.g., biblioteca, agência bancária, loja de roupas presencial ou virtual)

Um **clã** do MMORPG Perfect World é um grupo de jogadores unidos com o objetivo de jogar em conjunto e conquistar territórios. No jogo, há uma instância chamada Domínio das Dez Dimensões. Nesta instância, o clã se une para vencer 10 salas e receber premiações por elas.

O clã conta com um grupo de **Staffs** responsáveis pela organização da realização da instância Domínio das Dez Dimensões. A instância conta com uma **premiação** a cada sala completa, o que resulta em diversos **grupos de itens** a serem sorteados entre os **membros do clã**. Este sorteio é realizado pelos membros da Staff. Cada membro do clã pode manifestar interesse em participar do sorteio, sendo então adicionados à lista de **participantes do sorteio**.

Processo de trabalho

18. **Processo de trabalho** (identifique e descreva o processo de trabalho que será foco da solução proposta) (e.g., processo de controle de usuários, processo de controle de estoque, processo de venda, ...)

O processo de sorteio da premiação acontece com a manifestação de interesse dos membros pelos itens ganhos pelo clã. Os itens a serem sorteados são separados em **grupos**, cada um com um tipo de item.

Para cada grupo de itens é feito um sorteio. Este sorteio é realizado entre os membros do clã que manifestaram interesse em tal item e participaram da realização da instância. Contudo, há uma dificuldade em controlar quais membros **têm interesse** em quais itens, e quais membros **têm direito** a participar do sorteio.

Problema a ser resolvido

19. **Problema a ser resolvido** (descreva o problema) (e.g., “Controlar quais usuários estão com livros emprestados”; “Identificar quais usuários estão com entregas em atraso”)

O objetivo deste sistema será controlar quais membros têm interesse em quais itens, e quais membros têm direito a participar desses sorteios.

Tabelas de requisitos

Requisitos de software (**especificar 3 requisitos**)

20. **Requisitos de software** a serem implementados e **forma de implementação** (mínimo de 3 requisitos específicos para “**problema a ser resolvido**”) (criar tabela) (e.g., Especificar formato de dados ⇔ Criação de ontologia; Fazer cadastro de usuários ⇔ Cadastro de indivíduos usando Protégé OWL ou Protégé Web; Consultar usuários em atraso ⇔ Fazer consulta SPARQL)

ID	Requisito	Forma de implementação	Descrição
I001	Listar informações de membros do clã	Consulta SPARQL	A consulta deve mostrar o nick do jogador e a classe que pertence
I002	Listar participantes	Consulta SPARQL	Deve consultar quais

	do sorteio		membros do clã têm interesse no item sorteado e têm direito a participar do sorteio
I003	Listar sorteados	Consulta SPARQL	Deve consultar quais são os membros sorteados

Modelagem de dados

21. **Modelagem de dados** (identificar em quais ontologias/classes/propriedades cada um dos requisitos irão impactar) (criar tabela à parte ou incorporar tabela de **Requisitos de software**)

ID	Modelos de dados (ontologias/classes/propriedades/relacionamentos)
I001	Membros: Staffs; Soldados Staffs: nick; classe; interesse; participação Soldados: nick; classe; interesse; participação Clã/Instância tem Staffs Staffs organizam Instância
I002	Participantes: Membros Participantes: nick; classe; interesse; participação
I003	Sorteados: data_do_sorteito; item

Usuários

22. **Perfil de usuários** (criar tabela de funcionalidades por usuário) (identificar perfis de usuários do sistema e as várias funcionalidades que cada um poderá usar) (e.g., Administrador ⇔ Criação ontologia OWL com Protégé, Gerente, Estagiário ⇔ Consultar, ...)

Perfil do usuário	Funcionalidade
Staffs	Consultas I001 Cadastro I001 Consulta I002 Cadastro I002 Consultas I003 Cadastro I003
Soldados	Consultas I002 Consultas I003

Requisitos de interface

23. **Requisitos de interface** (identificar como será a interface para acesso às funcionalidades) (criar tabela)
(associar Requisitos de software ⇔ Usuário(s) ⇔ Descrição de requisitos de interface com identificação de ambiente) (identificar parâmetros de entrada e saída)

ID	Usuário	Descrição requisito de interface
I001	Staff	Receber o nick do Membro Devolver interesse e participação do Membro
I002	Staff	Receber o nick do Membro Devolver interesse e participação do Membro
I002	Soldado	Receber o nick do Membro Devolver interesse e participação do Membro
I003	Soldado	Receber o nick do Membro e devolver data_do Sorteio e item