

Modelo para Representação do Conhecimento em Pensamento Computacional

Ândrea V. Garcez¹, Braz A. S. Junior¹, Ana M. Pernas¹,
Simone A. da C. Cavalheiro¹, Luciana Foss¹

¹Centro de Desenvolvimento Tecnológico – Programa de Pós Graduação em Computação – Universidade Federal de Pelotas (UFPEL)

Abstract. *Computational Thinking (CT) is worked on by professionals from different areas, who seek to develop CT concepts and skills in different levels of education. However, its use results in divergences regarding the standardization of tasks, concepts and skills. A possible solution is to model the CT concepts and make them available as linked open data. Thus, an ontology-based knowledge model was created to represent and integrate multiple authors' perspectives on CT. The model was tested in typical scenarios, proving to be capable of representing different authors' points of view and linking resources of different natures according to their purposes. The model developed was made available openly for use by the community.*

Resumo. *O Pensamento Computacional (PC) é trabalhado por profissionais de diferentes áreas, que buscam desenvolver os conceitos e habilidades do PC nos diversos níveis de ensino. Entretanto, seu uso resulta em divergências quanto à padronização de tarefas, conceitos e habilidades. Uma possível solução é modelar os conceitos do PC e disponibilizá-los como dados abertos ligados. Desta forma, foi criado um modelo de conhecimento baseado em ontologias para representar e integrar múltiplas perspectivas de autores. O modelo foi testado em cenários típicos, mostrando-se capaz de representar pontos de vista de diferentes autores e de vincular recursos de diferentes naturezas de acordo com suas finalidades. O modelo desenvolvido foi disponibilizado de forma aberta para uso pela comunidade.*

1. Introdução

Nos últimos anos, iniciativas têm visado introduzir conceitos relacionados à programação e à robótica na educação básica [Blikstein 2018]. O Pensamento Computacional (PC) é um dos conhecimentos introduzidos mediante essas iniciativas, o qual se fundamenta em conceitos de computação. Segundo [Denning and Tedre 2019], PC pode ser considerado como conjuntos de habilidades que auxiliam no processo de resolução de problemas.

O PC é trabalhado em diversos níveis de ensino por profissionais que buscam desenvolver os conceitos e habilidades relacionados ao tema desde os primeiros anos de ensino. Entretanto, por esta ser uma área abrangente, na qual autores muitas vezes adotam abordagens distintas para tratar tópicos em comum, observa-se divergências quanto à padronização de tarefas, conceitos e habilidades. Por este motivo, observou-se no PC uma área oportuna para disponibilização dos dados de forma aberta e conectada. Ontologias mostram-se como uma forma de solução para representar conhecimento de forma compartilhada, permitindo integração e interoperação interpretável por máquina.

Nesta perspectiva, foi proposto um modelo semântico para padronização e representação do conhecimento em PC, o qual pode ser acessado pelas diferentes pesquisas, tornando possível o compartilhamento e reuso de informações. Nele são definidos conceitos para classificação de recursos educacionais, relacionado-os a alvos de aprendizagem. Com isso, busca-se incorporar conceitos de múltiplos autores, ao invés de definir arbitrariamente o que é PC. O modelo semântico é formado pela integração de 3 ontologias de topo, que representam abordagens de 3 autores diferentes em PC, e 1 ontologia de domínio, para modelagem dos recursos educacionais a serem classificados.

Um dos objetivos principais dessa pesquisa é tornar os conceitos modelados disponíveis e acessíveis de forma aberta e conectada. Tendo isto em vista, a ontologia integrada foi publicada em um *endpoint*¹ para acesso geral. Para verificar sua correção, foram instanciados dados de diferentes cenários, típicos do PC, e consultados diretamente no *endpoint*. Alguns resultados deste trabalho são apresentados neste artigo.

O texto está organizado da seguinte forma: na seção 2 são apresentadas as abordagens do PC adotadas na pesquisa desenvolvida; na seção 3 são definidos os conceitos em ontologias utilizados, assim como a base para o modelo da ontologia integrada; na seção 4 são apresentados os trabalhos relacionados; o modelo conceitual desenvolvido é apresentado na seção 5 e os cenários e testes na ontologia publicada estão apresentados na seção 6; conclusões e trabalhos futuros são descritos na seção 7.

2. Abordagens em Pensamento Computacional

Traçar definições na área de PC é um desafio visto ser um campo abrangente e com atuação de profissionais de áreas distintas. Em uma revisão da literatura observou-se que os trabalhos de Jeannet Wing, Karen Brennan e Mitchel Resnick são principalmente usados como base [Silva Junior et al. 2021]. Estes autores apresentam publicações com propósitos e naturezas distintas - enquanto Jeannet Wing tem influentes pontos de vista, Karen Brennan e Michel Resnick apresentam conceitos sólidos e referenciados em PC.

2.1. Conceitualização em PC com base em Jeannet Wing

Jeannette Wing foi uma das principais responsáveis pela disseminação e popularização do termo PC, em [Wing 2006] ela resume PC como um conjunto de habilidades utilizadas para resolução de problemas.

De acordo com [Wing 2006] e [Wing 2011], problemas complexos podem ser resolvidos usando decomposição e abstração. Métodos e modelos computacionais possibilitam a resolução de problemas e suas complexidades explicam a escolha de métodos e modelos computacionais. O PC é uma abordagem que busca a eficácia com eficiência, considerando a complexidade da tarefa envolvida. Modelar sistemas, resolver problemas e compreender o comportamento humano são tarefas que envolvem conceitos do PC.

2.2. Conceitualização em PC com base em Brennan e Resnick

Segundo [Brennan and Resnick 2012], as competências de PC são desenvolvidas a partir de três vertentes: **conceitos computacionais**, **práticas computacionais** e **perspectivas computacionais**. Uma atividade ou tarefa é expressada por uma série de etapas, e uma

¹<http://ontopact.ddns.net:8890>

sequência de instruções especifica uma ação que deverá ser realizada. O entendimento dos **conceitos computacionais** são necessários para que as pessoas saibam como realizar essas instruções, visando a reprodução de um determinado comportamento. **Sequências** são usadas para orquestrar as instruções. **Paralelismo** permite que mais de uma instrução seja realizada ao mesmo tempo. Para se executar uma sequência diversas vezes, são usados os **loops**. Já os **eventos** possibilitam que diferentes ações sejam tomadas de acordo com cada situação. Os condicionais permitem que sejam tomadas decisões. **Operadores** possibilitam manipulações numéricas e de texto. Por fim, os **dados** nada mais são do que variáveis, as quais são utilizadas para armazenar, recuperar e atualizar valores.

Práticas computacionais desenvolvem competências para representar os conceitos computacionais, sendo elas divididas em: **incremental e iterativa, teste e depuração, reutilização e remixagem, abstração e modularização**. Observando [Brennan and Resnick 2012], o desenvolvimento de um projeto é um processo adaptativo e incremental, ou seja, ele pode mudar de acordo com a abordagem utilizada para resolução. Testes e depuração são necessários para validar o que é pensado, pois muitas vezes o pensamento não prevê todas as possibilidades e por vezes se esquece algo importante que faz com que todo o desenvolvimento esteja errado. A reutilização e remixagem são necessárias para evitar que tenhamos que pensar sobre o mesmo problema diversas vezes e assim permitindo que seja utilizado o que foi desenvolvido e validado anteriormente. Já a modularização e a abstração possibilitam que a reutilização e remixagem sejam mais simples, pois permite que pessoas leigas abstraíam o conhecimento que foi usado para resolver algum problema, e o usem dentro do seu projeto, sem precisar entender como o mesmo foi desenvolvido.

As **perspectivas computacionais** desenvolvem como se **expressar**, como se **conectar** e como **questionar**. De acordo com [Brennan and Resnick 2012], o desenvolvimento da competência de expressão modifica como as pessoas pensam, passando a ver o PC como algo que possa ser consumido e usado para a resolução de problemas. A conexão com outras pessoas e com outros projetos desenvolve a criatividade, de onde surgem novas ideias. E o questionamento é utilizado para ir além das limitações do que é dado como certo, sendo utilizado para melhorar o que já existe.

2.3. Conceitualização em PC com base em Silva Junior

Em [Silva Junior 2020] é proposto um modelo que comporte a maior parte das conceitualizações abordadas pelos principais autores em PC. Os autores dividem PC em 6 linhas, cada uma abordando diversos conceitos e eventualmente se cruzando com outras: **abstração**, **decomposição**, **algoritmo**, **dados**, **avaliação** e **automação**. A linha de **abstração** leva a conceitos como modelagem, generalização e reconhecimento de padrões. A linha de **decomposição** aborda entre outros o reuso, a recursão e a composição. Já **algoritmo** abrange conceitos como paralelismo, fluxo de controle e funções. Coleta, visualização e estruturas de dados são exemplos de pontos que a linha de **dados** considera. Assim como **avaliação** trata otimização, teste e depuração. Por fim, na linha de **automação** está incluso desenvolvimento de software, hardware e simulações. Diversos outros conceitos são distribuídos pelas linhas e é ressaltado pelos autores que o modelo não é fechado. Isto é, o PC pode abordar outros pontos que não estão no modelo e outras linhas podem ser sugeridas.

O modelo foi criado a partir de uma revisão sistemática da literatura, capturando

os termos que são relacionados ao PC em 28 artigos frequentemente citados para definir PC [Silva Junior 2020]. A proposta era alcançar um modelo abrangente, que fizesse referência ao grande número de conceitos que são trabalhados dentro do PC. Os autores propõem as linhas e os pontos por onde elas passam como uma alternativa para evitar hierarquias e discordâncias quanto ao pertencimento de cada termo a cada categoria. Os pontos indicam somente que há alguma relação do conceito com a sua linha, não que ele pertence ou está restrito àquilo. É enfatizado também que as linhas se cruzam em diversos pontos e os conceitos são bastante conectados, interagindo uns com os outros.

3. Ontologias, Web Semântica e LOD

Ontologias consistem de vocabulários, ou seja, conjuntos de termos, que visam descrever adequadamente o conhecimento retido por determinados grupos, mantendo a expressividade dentro dos contextos em que são utilizados. Seu intuito é reduzir a ambiguidade na interpretação dos termos através da explicitação das relações entre os conceitos que o formam, sendo interpretável/processável por máquinas [Isotani and Bittencourt 2015].

Por serem estruturas de dados usadas para representar qualquer coisa existente, são adotadas em sistemas da Web Semântica como modelo de representação de conhecimento, interligando páginas e seus respectivos termos. A Web Semântica pode ser vista como uma implementação prática das ontologias (neste caso, tratadas como ontologias leves) e os Dados Abertos Conectados (Linked Open Data - LOD) como uma operacionalização da Web Semântica.

Dados conectados são dados ligados entre si, ou seja, representam e descrevem as conexões entre recursos presentes na web. De acordo com [Berners-Lee 2006], visto que a web mantém *hiperlinks* para interligar documentos da web, os dados conectados possuem 4 princípios: usar URIs para nomear coisas; usar URIs HTTP (Hypertext Transfer Protocol) para permitir que as pessoas procurem o que desejam; fornecer informações úteis por meio das URIs usando padrões RDF (Resource Description Framework) e SPARQL; incluir *links* para outros URIs, possibilitando explorar mais dados.

Dados são extraídos de inúmeras fontes e, por isso, acabam surgindo problemas de acesso e interoperação de dados. Como solução surgiu a LOD Cloud [Uyi Idehen 2019], criada com base em padrões abertos de acesso a dados e sistemas de gerenciamento de banco de dados que armazenam, indexam e realizam consultas a dados estruturados e *endpoints*, usados para acesso aos dados publicados no formato de LOD.

A proposta deste trabalho é compartilhar o conhecimento de autores do PC seguindo os princípios de uma LOD Cloud. Neste caso, os vocabulários foram representados por meio de ontologias em linguagem OWL (Web Ontology Language), as quais foram armazenadas no sistema gerenciador de banco de dados OpenLink Virtuoso² e disponibilizadas em um *endpoint*, podendo ser consultadas usando linguagem SPARQL.

4. Trabalhos Correlatos

Contrariamente ao esperado na pesquisa, não foram encontradas muitas ontologias específicas para representação de termos do PC. O trabalho encontrado mais próximo a proposta desenvolvida foi o Micas [Azevedo et al. 2019], uma plataforma web de apoio

²<https://virtuoso.openlinksw.com/>

aos professores no ensino de computação, através da coleta e classificação de recursos de aprendizagem a serem usados em aulas de informática.

A plataforma Micas permite que sejam inseridos quaisquer tipos de recursos de aprendizagem aplicados por professores no ensino de informática. Ela provê uma ontologia que é utilizada para categorizar e pesquisar recursos de aprendizagem por área e por ano de ensino. A ideia dos autores é auxiliar os professores na busca por recursos a serem aplicados em sala de aula para ensino de PC de acordo com o nível de ensino. A proposta do Micas, apesar de contar com uma ontologia disponibilizada através de uma plataforma web para pesquisa de recursos de aprendizagem, tem um propósito diferente do exposto neste trabalho, no qual objetiva-se a padronização e integração dos conceitos de PC disponibilizada e acessível de forma aberta na web.

Já relacionado a LOD na educação, [Bandeira et al. 2015] discute o potencial de dados abertos conectados para aperfeiçoar a educação através da produção, do compartilhamento e do reuso de dados. Segundo o autor, existem diversas iniciativas, entre elas a *LinkedEducation.org*³, que revelam que o uso de dados abertos conectados pode fomentar atividades relacionadas à educação e viabilizar a criação de bases de conhecimento que irão possibilitar uma melhor gestão e desenvolvimento educacional.

Em [Alcantara et al. 2015] são elencados os principais desafios quanto ao consumo e publicação de dados abertos conectados na educação brasileira. Os autores buscam estabelecer discussões que proponham soluções para esses desafios, uma vez que muitas bases apresentam limitações que dificultam o enriquecimento dos dados e conexão com outras bases de dados.

5. Modelo de Representação do Conhecimento em PC

De acordo com seu propósito, ontologias podem ser categorizadas da seguinte forma [Guarino 1998]:

- de topo:** usadas para descrever conceitos gerais como objetos, tempo e eventos;
- de domínio:** descrevem vocabulários relacionados a determinados tópicos. Neste trabalho, a ontologia de domínio estabelece conceitos básicos para que seja possível classificar, organizar e relacionar recursos ou tarefas de PC;
- de tarefa:** aplica o conhecimento para resolver problemas em inúmeras circunstâncias, sem se importar em qual domínio está;
- de aplicação:** são usadas para aplicar conceitos de domínio e tarefas, ou seja, são funções que as entidades realizam quando desempenham determinada tarefa.

Dado o objetivo deste trabalho de integrar perspectivas e visões de diferentes autores, foram definidas três ontologias de topo, com visões dos três autores do PC descritos na seção 2. Elas são usadas no modelo de representação proposto neste artigo, mas estão descritas em detalhe em [Silva Junior et al. 2021]. Como entende-se por necessário que outras visões, de outros autores, possam ser incorporadas futuramente ao modelo, agrega-se uma ontologia de domínio, a qual importa os conceitos de alto nível das ontologias de topo definidas. A criação do vocabulário da ontologia de domínio partiu da premissa de definição de conceitos do domínio educacional. A ontologia é descrita em detalhes em [Silva Junior et al. 2021]. Entretanto, para facilitar a compreensão neste trabalho, é apresentada na Figura 1.

³<http://www.linkededucation.org/>

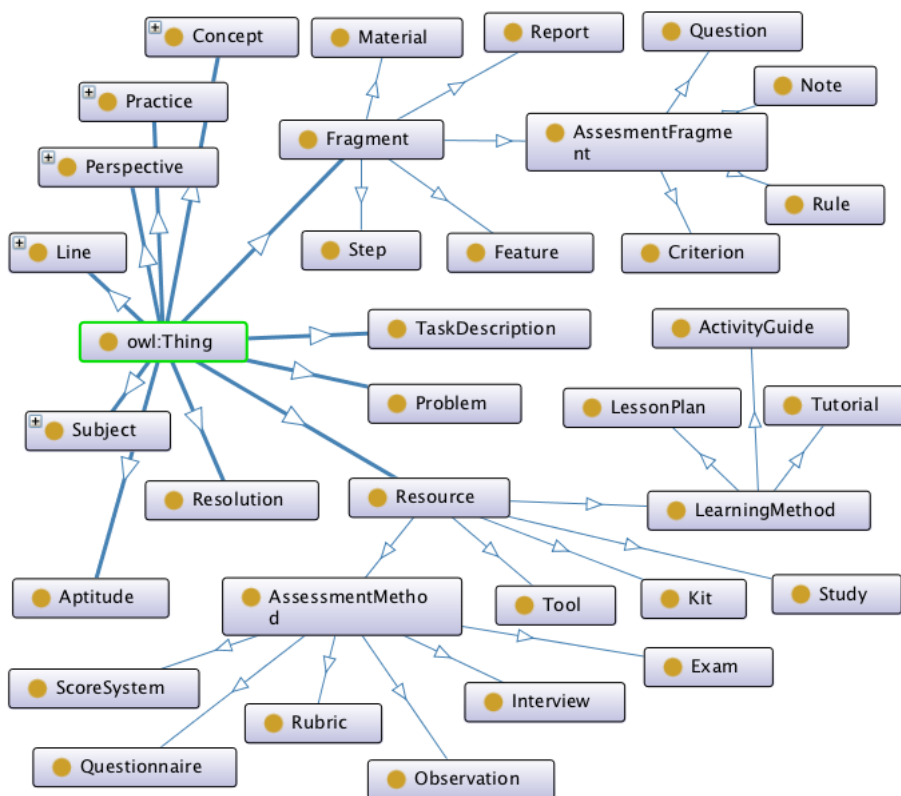


Figura 1. Ontologia de domínio

O conceito principal criado na ontologia do domínio educacional é o conceito **Subject**, que pode ser interpretado pelo conceito de **alvo de aprendizagem** do PC. Ele se comporta como um conceito de mais alto nível, como uma camada interna ao modelo, pois liga os recursos educacionais instanciados na ontologia de domínio aos conceitos das ontologias de topo definidas. A Figura 2 mostra a conexão entre as ontologias de topo e a ontologia de domínio. Esta integração entre as conceitualizações de PC e o domínio educacional forma a camada de representação do conhecimento proposta neste trabalho, e que possibilita a inclusão de novos conhecimentos de topo por meio da importação de novas ontologias, de acordo com os princípios de LOD.

Na Figura 2 pode-se verificar a ontologia de domínio apenas com as partes que se relacionam com as ontologias de topo. As classes **conceito**, **perspectiva**, **prática** que estão dentro do retângulo preto são as classes oriundas da importação da ontologia de topo de Brennan, a classe **linha** e suas subclasses que estão com contorno em vermelho aparecem devido a importação da ontologia de topo de Silva Junior. A ontologia de topo da autora Wing, por não fornecer uma hierarquia organizada em alvos de aprendizagem, é representada no domínio diretamente por subclasses dentro da classe alvos de aprendizagem. Na Figura 2 essas subclasses são as que possuem um retângulo azul em volta.

6. Disponibilização do Modelo e Cenário de Testes

Um dos objetivos da pesquisa é tornar o modelo proposto disponível e acessível seguindo os padrões definidos por LOD. Desta forma, foram definidas as ontologias usando o software Protégé e exportada no modelo RDF. O código correspondente, em formato RDF,

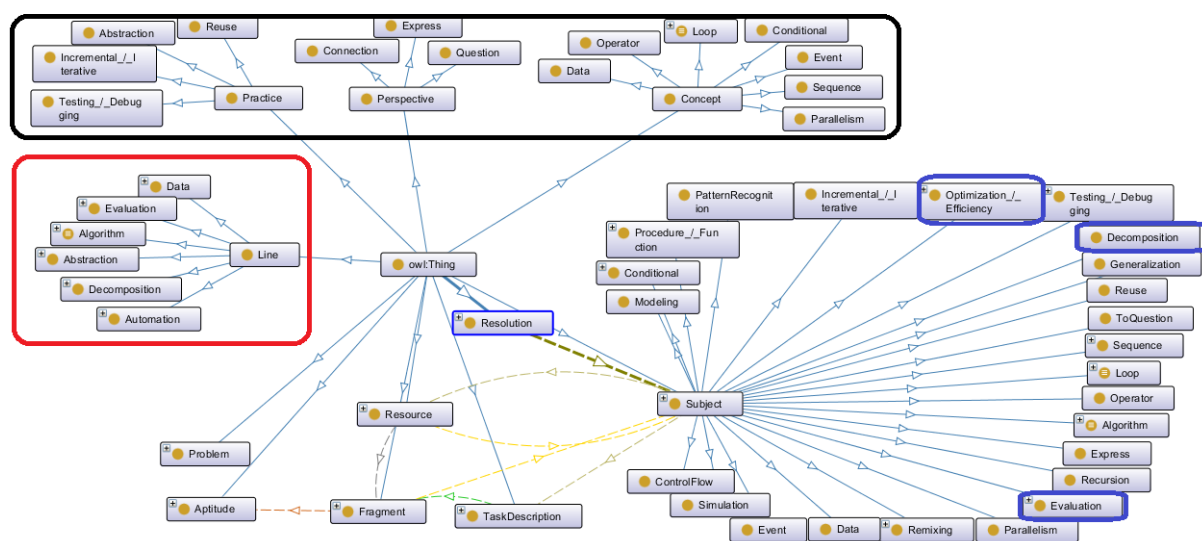


Figura 2. Camada de representação do conhecimento em PC

foi importado para o OpenLink Virtuoso via *quad store upload*⁴.

Para realizar consultas no *endpoint* público definido, basta acessar o IP:PORTA/sparql que, no caso deste projeto, ficou como <http://ontopact.ddns.net:8890/sparql>. Neste momento, é apresentada uma página que permite a realização de consultas SPARQL, então no campo *default data set name (graph IRI)* é necessário informar o *endpoint* público, o qual foi definido no campo *named graph IRI* do OpenLink Virtuoso.

Para avaliar o modelo proposto, foram criados quatro cenários com recursos de diferentes naturezas. Os cenários foram os seguintes: teste de pensamento computacional (*computational thinking test - CTt*)⁵, ciência da computação desplugada (*computer science unplugged - CSU*) [Bell et al. 2009], Scratch⁶ e Jogo A Última Árvore.

Os 3 primeiros cenários estão descritos em [Silva Junior et al. 2021]. O cenário A Última Árvore consiste de um jogo modelado como uma gramática de grafos para desenvolver habilidades do PC. Ele foi proposto inicialmente como um jogo físico de tabuleiro, de estratégia por turno, que convida 4 jogadores a reflorestar uma selva recentemente desmatada a partir de sua última árvore [Silva Junior et al. 2017]. O jogo gira em torno da transformação de recursos (sementes, plantas, árvores e frutas) utilizando regras de transformação de grafos, onde os jogadores precisam analisar o tabuleiro a fim de identificar padrões nos grafos para poder aplicá-las. O jogador deve traçar uma estratégia para sequenciar a aplicação dessas regras de forma eficiente, para conseguir alcançar seu objetivo antes dos outros competidores.

Em seguida, foi proposta uma versão digital do jogo [Silva Junior et al. 2019],

⁴Função do Virtuoso que permite carregar arquivos OWL e RDF para realização de consultas SPARQL

⁵Teste de pensamento computacional, encontrado em: https://docs.google.com/forms/d/e/1FAIpQLSdPdSj_zVUhIhG4S3bCH6zXSHZoHHbv6OsmCF9drmbDpfBy_Q/viewform?fbzx=1710660382635002705

⁶<https://scratch.mit.edu/>

onde há um sistema de captura automática e tratamento de dados sobre o jogo para permitir uma avaliação mais precisa. Por exemplo, para aplicar determinada regra, o jogador precisa identificar um padrão de recursos que representa uma condição para aquela regra. Esta condição é apresentada como um grafo e cada um dos elementos dele precisa ser mapeado em um elemento correspondente no tabuleiro. É assim que o jogador indica o reconhecimento daquele padrão no jogo. No entanto, existem diversas formas de erro possíveis nesse mapeamento, como mapear elementos de tipos diferentes (e.g. mapear uma árvore em um animal). A versão digital registra todos os mapeamentos, indicando os erros cometidos, seus tipos e detalhes como o tempo levado para completá-lo. Utilizando esses dados e comparando-os com os de outros jogadores, o avaliador pode ter uma visualização mais precisa das habilidades e eventuais carências do PC de cada jogador.

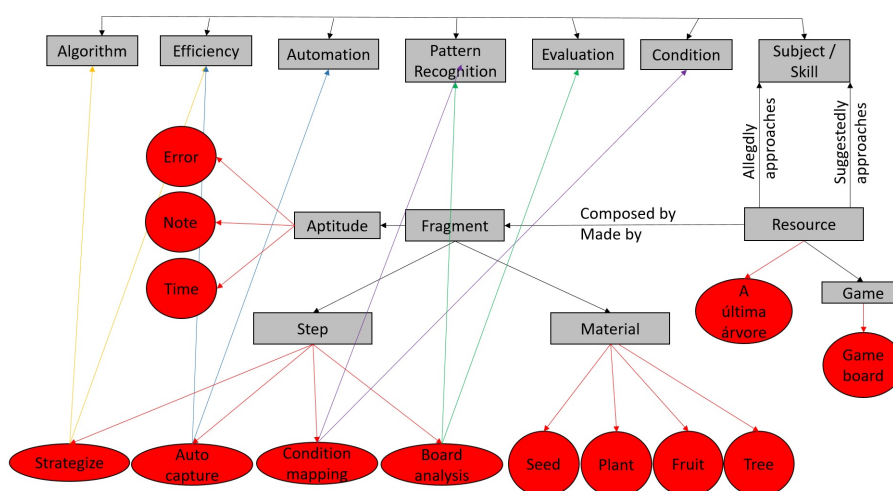


Figura 3. Cenário de representação do jogo A Última Árvore

Na Figura 3 é apresentado o cenário do Jogo A Última Árvore com recursos instanciados na ontologia de domínio desenvolvida. As classes são colocadas em **negrito**, as relações em *itálico* e as instâncias dentro de [colchetes].

No cenário mostrado na Figura 3, [A Última Árvore], que está representado na cor vermelha, é uma instância de **resource** e [Game Board] é um **resource** do tipo **Game**, indicando que A Última Árvore é um jogo de tabuleiro. Ele *aborda* **subject/skill** e é *composto por* quatro **Material** chamados de: **fruit**, **seed**, **tree** e **plant**. Também é *composto por* quatro passos. **Step** e **Material** são subclasses de **Fragment**. Cada um dos passos *requer* uma **aptitude** que pode ser [Note], [Time] ou [Error] cometidos.

Os passos desenvolvidos nesse cenário, são os seguintes: [Strategize] trata sobre a criação de estratégias para sequenciar a aplicação de regras de gramática de grafos, abordando **Algorithm** e **Efficiency**. [Auto capture] faz a captura automática permitindo avaliação mais precisa, abordando **Efficiency** e **Automation**. [Condition mapping] realiza o mapeamento das condições dos elementos para o tabuleiro, abordando **Condition** e **Pattern Recognition**. [Board analysis] consiste em analisar o tabuleiro para verificar quais regras deverão ser aplicadas, abordando **Pattern Recognition** e **Evaluation**.

Para teste do modelo foi realizada a consulta descrita na Figura 4, na qual é testado o cenário instanciado do Jogo A Última Árvore. Com a consulta SPARQL é possível

identificar que o recurso [A Última Árvore] explicado anteriormente está associado a um tipo de recurso e é capaz de desenvolver uma série de alvos de aprendizagem. Esse recurso é composto por uma série de fragmentos e um tipo de fragmento pode abordar um ou mais fragmentos. Cada fragmento é capaz de mapear quais alvos de aprendizagem ele desenvolve.

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontopact: <http://www.semanticweb.org/andre/ontologies/2021/2/untitled-ontology-8#>

SELECT ?resource ?resource_type ?fragment_type ?fragment ?subject_type
WHERE {
  ?fragment rdf:type ?fragment_type .
  ?fragment ontopact:makeByResource ?resource .
  {?fragment_type rdfs:subClassOf ontopact:Fragment .}
  ?resource ontopact:approachesSubject ?subject .
  ?resource rdf:type ?resource_type
  {?resource_type rdfs:subClassOf ontopact:Resource .}
  ?subject rdf:type ?subject_type
  {?subject_type rdfs:subClassOf ontopact:Subject .}
  FILTER(regex(str(?resource), "A_ultima_arvore"))
}
ORDER BY ASC(?resource) ASC(?fragment) ASC(?subject)
```

Figura 4. Consulta para o cenário A Última Árvore

O resultado da consulta da Figura 4 pode ser visto na Figura 3 em forma de grafo e na Figura 5 em forma de texto, resultado da consulta SPARQL realizada no endpoint da ontologia. O resultado da consulta valida o cenário descrito na Figura 3, mostrando quais são os alvos de aprendizagem que parte do recurso A Última Árvore desenvolve e quais fragmentos do tipo passo (step) e do tipo material fazem parte dele. Por questões de espaço, não são apresentados todos os recursos resultantes da consulta.

resource	resource_type	fragment_type	fragment	subject_type
A_ultima_arvore	Resource	Step	Auto_Capture	Automation
A_ultima_arvore	Resource	Step	Auto_Capture	Conditional
A_ultima_arvore	Resource	Step	Auto_Capture	Evaluation
A_ultima_arvore	Resource	Step	Auto_Capture	Optimization_/Efficiency
A_ultima_arvore	Resource	Step	Auto_Capture	PatternRecognition
A_ultima_arvore	Resource	Step	Board_Analysis	Algorithm
A_ultima_arvore	Resource	Step	Board_Analysis	Automation
A_ultima_arvore	Resource	Step	Board_Analysis	Conditional
A_ultima_arvore	Resource	Step	Board_Analysis	Evaluation
A_ultima_arvore	Resource	Step	Board_Analysis	Optimization_/Efficiency
A_ultima_arvore	Resource	Step	Board_Analysis	PatternRecognition
A_ultima_arvore	Resource	Step	Condition_Mapping	Algorithm
A_ultima_arvore	Resource	Step	Condition_Mapping	Automation
A_ultima_arvore	Resource	Step	Condition_Mapping	Conditional
A_ultima_arvore	Resource	Step	Condition_Mapping	Evaluation
A_ultima_arvore	Resource	Step	Condition_Mapping	Optimization_/Efficiency
A_ultima_arvore	Resource	Step	Condition_Mapping	PatternRecognition
A_ultima_arvore	Resource	Material	Fruit	Algorithm
A_ultima_arvore	Resource	Material	Fruit	Automation

Figura 5. Resultado da consulta ao A Última Árvore

A Figura 6 mostra a mesma consulta executada para o A Última Árvore, mas com filtro para o cenário criado para o Scratch, no qual foi modelado o CATS!⁷, onde um aluno cria um aplicativo que coloca inúmeros gatos na tela. Similarmente à Figura 3, [Scratch]

⁷<https://projects.raspberrypi.org/en/projects/cats>

é um **recurso** do tipo **ferramenta** que *está relacionado* com outros **recursos**, no exemplo, *está relacionado* ao [CATS!], que é um **recurso de ensino** do tipo **plano de aula**.

No resultado da Figura 6 as cinco primeiras linhas, contornadas com um retângulo vermelho, mostram todos os cinco alvos de aprendizagem que o fragmento **CATS! clone cats** desenvolve. Ele é um fragmento do tipo **passo (step)** e o fragmento **CATS! clone cats** pertence ao recurso **CATS!** que é um recurso do tipo **plano de ensino (lesson plan)**.

Os testes dos cenários mostram que o modelo proposto é capaz de mapear recursos que devem ser usados para o desenvolvimento de alvos de aprendizagem e também sua relação inversa. Permite, ainda, mapear quais alvos de aprendizagem um determinado fragmento pode desenvolver.

resource	resource_type	fragment_type	fragment	subject_type
CATS!	LessonPlan	Step	CATS!_-_clone_cats	Algorithm
CATS!	LessonPlan	Step	CATS!_-_clone_cats	Conditional
CATS!	LessonPlan	Step	CATS!_-_clone_cats	Evaluation
CATS!	LessonPlan	Step	CATS!_-_clone_cats	Procedure_/_Function
CATS!	LessonPlan	Step	CATS!_-_clone_cats	Loop
CATS!	LessonPlan	Step	CATS!_-_draw_lines	Algorithm
CATS!	LessonPlan	Step	CATS!_-_draw_lines	Conditional
CATS!	LessonPlan	Step	CATS!_-_draw_lines	Evaluation
CATS!	LessonPlan	Step	CATS!_-_draw_lines	Procedure_/_Function
CATS!	LessonPlan	Step	CATS!_-_draw_lines	Loop
CATS!	LessonPlan	Step	CATS!_-_make_the_cats_move	Algorithm
CATS!	LessonPlan	Step	CATS!_-_make_the_cats_move	Conditional

Figura 6. Resultado da consulta ao Scratch

7. Conclusões e Trabalhos Futuros

Sabendo do problema atual enfrentado pela comunidade de PC, relativo às dificuldades para definição consensual quanto aos significados dos termos e conjuntos de habilidades, neste trabalho foi proposto um modelo de conhecimento capaz de representar e integrar múltiplas perspectivas sobre PC. Este modelo foi desenvolvido como uma aplicação em dados abertos conectados, a fim de padronizar os conjuntos de habilidades de PC para que pesquisadores, professores e entusiastas da área possam trabalhar de forma coerente e convergente. O vocabulário gerado foi disponibilizado de forma aberta na web para que possa ser amplamente utilizada e extensível.

Como forma de teste do modelo desenvolvido, foram propostos cenários com visões de diferentes autores mostrando a expressividade, o alcance e a operação da ontologia de domínio. Com esses cenários o modelo revelou-se capaz de modelar diferentes visões de autores, bem como vincular recursos de diferentes naturezas de acordo com a finalidade e assunto abordado. Também é possível adicionar mais ontologias de topo com visões de outros autores e instanciar outros cenários.

Como trabalhos futuros percebe-se a necessidade de aumentar a descrição dos conceitos existentes a fim de utilizar a ontologia como forma de avaliar habilidades do PC, pois cada fragmento possui uma aptidão que é uma estimativa de sua dificuldade.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- Alcantara, W., Bandeira, J., Barbosa, A., Lima, A., Ávila, T., Bittercourt, I., and Isotani, S. (2015). Desafios no uso de dados abertos conectados na educação brasileira. In *Anais do IV Workshop de Desafios da Computação aplicada à Educação*, pages 11–20. SBC, SBC.
- Azevedo, A., Araújo, C., Henriques, P. R., Osório, A., Gomes, M., and Valente, A. (2019). Micas, a web platform to support teachers of computing at school. *Challenges*, pages 625–641.
- Bandeira, J., Ávila, T., Alcantara, W., Sobrinho, A., Bittencourt, I. I., and Isotani, S. (2015). Dados abertos conectados para a educação. *Jornada de Atualização em Informática na Educação*, 4(1):47–69.
- Bell, T., Alexander, J., Freeman, I., and Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1):20–29.
- Berners-Lee, T. (2006). Linked data. <https://www.w3.org/DesignIssues/LinkedData>. [Online; acessado em 28-novembro-2020].
- Blikstein, P. (2018). Maker movement in education: History and prospects. In *Handbook of Technology Education*, pages 419–437. New York City.
- Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, volume 1, page 25. American Educational Research Association.
- Denning, P. and Tedre, M. (2019). *Computational Thinking*. Cambridge, Massachusetts, United States: MIT Press.
- Guarino, N. (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, Amsterdam.
- Isotani, S. and Bittencourt, I. I. (2015). *Dados Abertos Conectados: Em busca da Web do Conhecimento*. Novatec Editora, São Paulo.
- Silva Junior, B. A. d. (2020). Ggascet: bringing formal methods to the computational thinking. Master's thesis, Universidade Federal de Pelotas.
- Silva Junior, B. A. d., Cavalheiro, S. A. d. C., and Foss, L. (2017). A última árvore: exercitando o pensamento computacional por meio de um jogo educacional baseado em gramática de grafos. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, pages 735–744.
- Silva Junior, B. A. d., Cavalheiro, S. A. d. C., and Foss, L. (2019). Revisitando um jogo educacional para desenvolver o pensamento computacional com gramática de grafos. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 863.
- Silva Junior, B. A. d., Cavalheiro, S. A. d. C., Foss, L., Fleischmann, A. M. P., and Garcez, A. V. (2021). Pact: A citizen science project for computing education. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–9. IEEE, IEEE.

Uyi Idehen, K. (2019). What is the linked open data cloud, and why is it important? <https://medium.com/virtuoso-blog/what-is-the-linked-open-data-cloud-and-why-is-it-important-1901a7cb7b1f>. [Online; acessado em 1-novembro-2021].

Wing, J. (2011). Research notebook: Computational thinking — what and why. *The link magazine*, 6:20–23.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.