

Revista Comunicações em Informática



Um Mapeamento Sistemático sobre Sistemas que Envolvem IoT e Deep Learning Conjuntamente

Janyelson V. L. de Oliveira, João L. F. Amorim, Matheus M. R. Praxedes

Centro de Informática – Universidade Federal da Paraíba

{janyelsonvictor, jony.lucas.br13, matheuspraxedes2014}@gmail.com

Resumo: Na última década, a Internet das Coisas (IoT) e *Deep Learning* receberam significativa atenção da indústria, bem como das universidades e dos centros de pesquisa. As principais razões por trás desse interesse está no fato que IoT permite conectar diversos dispositivos ao nosso redor por meio da Internet - criando ambientes inteligentes - e *Deep Learning* mostrou-se ter bons resultados em uma série de aplicações, em comparação com as abordagens tradicionais, principalmente, na classificação de imagens. Neste trabalho realizamos um mapeamento sistemático de literatura com o objetivo de identificar como IoT e *Deep Learning* podem funcionar conjuntamente em diversos sistemas e aplicações. Os resultados apontaram que a união dessas duas abordagens é útil para a solução de vários problemas práticos.

Palavras-chave: *Deep Learning*; IoT; Mapeamento sistemático.

1. Introdução

Sistemas IoT (em inglês, *Internet of Things*) e *Deep Learning* (em português, Aprendizado Profundo) são termos comuns quando pensamos em tecnologias modernas que utilizam comunicação entre componentes eletrônicos e aprendizagem de máquina, respectivamente. Nos últimos anos, vêm surgindo bastante interesse tanto no desenvolvimento e uso de sistemas baseados em IoT quanto em aplicar técnicas de redes neurais artificiais (ANNs), especificamente *Deep Learning*, para aprendizagem e predição com o intuito de resolver algum tipo de problema [25,26].

Não há como medir a extensão dos sistemas IoT, pois o termo IoT engloba tecnologias compostas por uma estrutura que permite a identificação e comunicação por meio de radiofrequência (RFID), comunicação por meio da Internet, por protocolos e tecnologias de rede e traz desafios adicionais como armazenamento e compartilhamento de dados [25].

A ideia básica por trás da IoT é conectar todas as coisas ao nosso redor utilizando a Internet, fazendo com que se tenha um ambiente inteligente, onde os diversos dispositivos podem se comunicar uns com os outros de forma semelhante à comunicação entre humanos [8].

Deep Learning é imensamente utilizado em tecnologias que envolvem predição a partir de dados coletados, permitindo o desenvolvimento de tecnologias como filtros na Internet (e.g. filtro de *spams*), recomendações de produtos, entre outros [26]. Os estudos e o desenvolvimento da aprendizagem profunda possibilitou grandes progressos também nas áreas de reconhecimento de imagens e processamento de voz. As redes neurais de aprendizado profundo são superiores a outros métodos tradicionais de reconhecimento de imagem e tem feito grandes progressos no processamento de dados massivos e na computação distribuída [5].

Neste artigo é realizado um mapeamento sistemático com intuito de encontrar artigos relacionados a sistemas IoT e *Deep Learning*. O objetivo desse mapeamento é entender como funcionam (métodos, tecnologias, arquitetura, integração, comunicação e princípios) os sistemas que utilizam *Deep Learning* e IoT conjuntamente, além de buscar as principais aplicações (contextos de uso) desses sistemas.

2. Metodologia

O procedimento metodológico escolhido foi o mapeamento sistemático, pois utiliza uma metodologia bem definida que permite identificar e analisar tópicos de pesquisa em diferentes áreas do conhecimento, inclusive em áreas da computação.

O mapeamento sistemático foi feito sobre artigos que envolvem estudos e/ou desenvolvimento de sistemas baseados em IoT e *Deep Learning* conjuntamente. Diante disso, foram criadas questões que atendessem aos principais objetivos da pesquisa, são eles: descobrir como funcionam os sistemas que utilizam *Deep Learning* e IoT conjuntamente (principais arquiteturas, tecnologias, princípios e requisitos dos sistemas); e quais as principais aplicações e contextos de uso desses sistemas, como mencionado na seção anterior. O mapeamento sistemático foi executado, em uma visão mais geral, com os seguintes passos:

1. Conhecido o tópico/tema de pesquisa, elaborar questões de pesquisa que deverão ser respondidas ao longo de todo o mapeamento;
2. Elaborar uma *string de busca* (ou chave de busca) para o tema proposto;
3. Escolher os motores de buscas pelo qual a *string* será executada e os artigos serão pesquisados;
4. Definir quais serão os critérios de inclusão e exclusão para a seleção dos artigos; e
5. Selecionar nos motores de busca escolhidos, ao longo de várias etapas consecutivas, quais artigos respondem às questões de pesquisa para o tema proposto e que também atendam aos critérios de inclusão e exclusão.

2.1 Questões de Pesquisa

A presente subseção mostra quais foram as questões de pesquisa utilizadas ao longo do mapeamento, são elas:

Questão 1: Como a IoT pode contribuir nos sistemas e aplicações que utilizam *Deep Learning* e vice-versa?

Questão 2: Quais os principais requisitos e princípios aplicados em sistemas que utilizam *Deep Learning* e IoT conjuntamente?

Questão 3: Qual(is) a(s) principal(is) arquitetura(s) empregada(s) em sistemas que utilizam *Deep Learning* e IoT conjuntamente?

Questão 4: Qual o contexto de uso desses sistemas? O objetivo é saber quais são os principais contextos de uso ou quais estudos sobre o tema já estão sendo realizados.

Questão 5: Quais exemplos de aplicações/sistemas existentes?

2.2. Condução da Pesquisa

Após a definição das questões de pesquisa, a busca e seleção dos artigos foram feitas utilizando dois dos principais motores de busca (ou base de dados) que possuem artigos na área de computação ou informática e que apresentaram melhores resultados de artigos relacionados com *Deep Learning* e sistemas IoT, são eles: Scopus e Springer.

A *string de busca* foi elaborada com base nas questões de pesquisa e que apresentasse, ao mesmo tempo, um melhor resultado nos motores de buscas citados. Logo, a *string* escolhida foi: (“Internet of Things” OR IoT) AND (“Deep Learning”) - com as devidas modificações de sintaxe para que pudesse ser aceita nas duas bases.

2.3. Triagem dos Artigos

Para a seleção dos artigos relevantes para as questões de pesquisa, foram elaborados critérios de exclusão e de inclusão. Os critérios de inclusão escolhidos foram: publicações (entre 2014 e 2018); escritas em português e/ou inglês; com até 20 páginas; e que apresentassem contextos de uso. Entre os critérios de exclusão temos: publicações com até 5 páginas; não foram aceitas publicações duplicadas nem pagas ou não disponíveis nem revisões literárias.

Após a elaboração dos critérios, foram realizados os seguintes passos para a seleção dos artigos: o 1º passo foi selecionar todos os artigos resultantes da aplicação da *string* de busca nos motores de pesquisa de cada base de busca; o 2º passo foi realizar a leitura dos resumos dos artigos selecionados no passo anterior e realizar a primeira triagem de acordo com os critérios de inclusão e exclusão definidos; e, por último, o 3º passo consistiu na seleção dos artigos após a leitura da sua introdução e que também estivessem de acordo com os critérios de inclusão e exclusão citados.

3. Resultados e Discussão

Os resultados foram divididos em resultados gerais da pesquisa (englobando todos os artigos selecionados), seguido pelas respostas aos questionamentos propostos (também de acordo com cada artigo selecionado).

3.1. Resultados Gerais

No final, o mapeamento sistemático resultou em um total de 24 artigos. A maior quantidade de publicações ocorreu no ano de 2017 (com 15 artigos), seguido pelo ano de 2016 (com 5 artigos). Já nos anos de 2014 e 2018 foram publicados 1 e 3 artigos, respectivamente. No ano de 2015, não houveram publicações sobre o tema pesquisado nos motores de busca escolhidos.

3.2. Questão 1: Como a IoT pode contribuir nos sistemas e aplicações que utilizam Deep Learning e vice-versa ?

Por questões de simplificação e de melhor visualização, optamos, primeiramente, apresentar as contribuições de técnicas de *Deep Learning* para os sistemas IoT. Sendo assim, as contribuições encontradas nos artigos

decorrentes do mapeamento sistemático foram sumarizadas no Quadro 1.

Quadro 1: Contribuição de técnicas de *deep learning* para sistemas IoT.

Contribuição (<i>Deep Learning</i>)	#	Referências
Processamento e análise dos dados coletados	8	[3;4;5;16;20;21;22;24]
Extração de características	5	[2;14;20;21;23]
Reconhecimento/classificação de padrões visuais por imagens	4	[10;11;18;19]
Auxílio em tomadas de decisão	4	[1;2;4;9]
Melhorar o tráfego da rede	3	[6;7;17]
Reconhecimento/classificação de atributos	3	[13;20;23]
Predição de problemas	2	[16;22]
Sistemas baseados no conhecimento	2	[8;9]
Inferir situações/eventos com base nos dados coletados	1	[15]

Ao analisarmos o quadro, percebe-se que o processamento e análise dos dados coletados é a maior contribuição de *Deep Learning* para sistemas IoT (8/24 artigos), seguido pela extração de características (5/24 artigos) e reconhecimento/classificação de padrões visuais por imagens e auxílio em tomadas de decisão (ambos 4/24 artigos). Outras contribuições como reconhecimento/classificação de atributos e melhorias no tráfego da rede corresponderam, cada uma, com 3 dos 24 artigos resultantes do mapeamento.

3.3. Questão 2: Quais os principais requisitos e princípios aplicados em sistemas que utilizam Deep Learning e IoT conjuntamente?

Analisando os artigos escolhidos, percebemos que o requisito que mais aparece é a necessidade da existência de vários dados para treinamento dos sistemas que utilizam *Deep Learning*. Essa necessidade parece ser inerente a qualquer sistema que utiliza de alguma forma Aprendizado de Máquina. Além disso, para o processamento desses dados, se faz necessário máquinas com grande capacidade de processamento.

Outros requisitos e princípios foram encontrados, são eles: dinamicidade, eficiência, produtividade, conectividade, disponibilidade, inteligência, adaptabilidade, ergonomia, integração com outras tecnologias de computação, comunicação, estabilidade, desempenho, confiabilidade, robustez, portabilidade entre plataformas, segurança, manutenibilidade, execução em tempo real e capacidade de generalização. Os sistemas propostos nos artigos utilizam boa parte desses princípios e requisitos em sua fase de estudo e desenvolvimento.

3.4. Questão 3: Qual(is) a(s) principal(is) arquitetura(s) empregada(s) em sistemas que utilizam Deep Learning e IoT conjuntamente?

O objetivo dessa questão é descobrir as principais arquiteturas empregadas em sistemas que utilizam *Deep Learning* e IoT conjuntamente. A partir da pesquisa

realizada, vimos que a arquitetura mais utilizada é a por camadas (12/24 artigos), onde 6/12 dos artigos são arquiteturas de 3 camadas e os 6/12 restantes são de 4 camadas. Em seguida, temos a arquitetura orientada ao conhecimento (3/24 artigos), que pode ser subdividida em: arquitetura por módulos de raciocínio, cíclica e de gerenciamento (ambas 1/24 artigos). Por fim, temos as arquiteturas baseadas na Internet (1/24 artigos) e em Sistemas Embarcados (1/24). O Quadro 2 apresenta o mapeamento das arquiteturas abordadas pelos artigos.

Quadro 2: Arquiteturas dos sistemas.

Arquitetura	Detalhamento	#	Referências
Por camadas	3 camadas: <ul style="list-style-type: none"> Física/Aplicação Redes Inteligência 	6	[4;6;13;16;19;23]
	4 camadas: <ul style="list-style-type: none"> Física Redes Inteligência Aplicação 	6	[1;10;11;21;22;24]
Orientado ao conhecimento	Módulos de raciocínio	1	[15]
	Cíclica	1	[9]
	Gerenciamento	1	[8]
Baseado na Internet	Redes de computadores	1	[14]
Sistemas embarcados	Dispositivos de baixa potência e baixo consumo de energia	1	[11]

3.5. Questão 4: Qual o contexto de uso desses sistemas?

A pesquisa mostrou que o principal contexto de uso é o reconhecimento de pessoas ou objetos em um ambiente (5/24 artigos), seguido pelo monitoramento de pacientes (4/24 artigos) e pelo monitoramento tanto de estruturas físicas como de uma rede em um sistema IoT (ambos 3/24 artigos). Além desses, existem outros, em menor quantidade, como o monitoramento de motoristas e gestão de pessoas (ambos 2/24 artigos) e o reconhecimento de contexto para os dados dos sensores de IoT e gerenciamento de ambientes (ambos 1/24 artigos). O Quadro 3 mostra o mapeamento dos contextos de uso dos sistemas retornados pela pesquisa.

3.6. Questão 5: Quais exemplos de aplicações/sistemas existentes?

Por fim, diferentemente da questão anterior, procuramos por exemplos de sistemas ou aplicações já em uso ou que estão em fase de estudo/desenvolvimento. Sendo assim, nosso estudo mostrou uma variedade de sistemas, entre eles, sistema de cuidados da saúde e para ambientes inteligentes (ambos 4/24 artigos), seguido por sistemas para verificar condições físicas de estruturas e de balanceamento de dados para IoT (ambos 3/24 artigos), além de sistemas de identificação de pessoas em lojas, de monitoramento do comportamento de motoristas e os de informação e conhecimento (ambos 2/24 dos artigos). O Quadro 4 mostra o mapeamento dos sistemas e aplicações já existentes retornados pela pesquisa.

Quadro 3: Contexto de uso dos sistemas.

Contexto	#	Referências
Reconhecimento de pessoas ou objetos em um ambiente.	5	[10;11;12;18;19]
Monitoramento de pacientes	4	[2;4;21;24]
Monitoramento de estruturas físicas	3	[16;20;22]
Monitoramento da rede de um sistema IoT.	3	[6;7;17]
Monitoramento de motoristas	2	[1;3]
Gestão do conhecimento	2	[8;9]
Reconhecimento de contexto para os dados dos sensores de IoT.	1	[15]
Gerenciamento de ambientes	1	[23]

Quadro 4: Exemplos de aplicações/sistemas.

Aplicações/Sistemas	#	Referências
Sistema de cuidados da saúde	4	[2;4;21;24]
Sistema para ambientes inteligentes	4	[2;10;19;23]
Sistema para verificar condições físicas de estruturas (geradores eólicos, estruturas civis e esgotos)	3	[16;20;22]
Sistema de balanceamento de dados para IoT	3	[6;7;17]
Sistema de identificação de pessoas em lojas	2	[5;18]
Sistema de monitoramento do comportamento de motoristas	2	[1;3]
Sistema de informação e conhecimento	2	[8;9]
Sistema para robôs inteligentes	1	[2]
Sistemas de orientação e informação de estacionamento	1	[11]
Sistema de reconhecimento, em tempo real, de objetos em uma imagem	1	[12]
Sistema de reconhecimento de atividades humanas	1	[13]
Sistema de raciocínio e inferência de situações	1	[15]

4. Conclusão

Avaliar todos os métodos e abordagens da Inteligência Artificial (IA) que podem ser trabalhados em conjunto com IoT, e vice-versa, é uma atividade bastante complexa, pois tanto a área de IA quanto a de IoT trabalham com diversas possibilidades e ferramentas. Nesse contexto, este trabalho realizou um mapeamento sistemático de literatura no intuito de conhecer como funcionam os sistemas que utilizam *Deep Learning* e IoT conjuntamente, limitando assim o escopo de estudo.

Conclui-se também que diversos sistemas e aplicações que utilizam IoT e *Deep Learning* conjuntamente já estão em fase de desenvolvimento ou de estudo e que a união dessas duas tecnologias é bem promissora para diversas atividades do cotidiano.

Bibliografia

- [1] Virmani, S. & Gite, S. (2017). Performance of Convolutional Neural Network and Recurrent Neural Network for anticipation of driver's conduct. In *Anais do IEEE*. DOI: 10.1109/ICCCNT.2017.8204039.
- [2] Chen, M., Herrera, F. & Hwang, K. (2017). Cognitive Computing: Human-Centered Computing with Cognitive Intelligence on Clouds. In *Anais do IEEE*. DOI: 10.1109/ACCESS.2018.2791469.
- [3] Streiffer, C., Raghavendra, R., Benson, T. & Mudhakar Srivatsa, M. (2017). DarNet: A Deep Learning Solution for Distracted Driving Detection. In *Anais da ACM*. DOI: 10.1145/3154448.3154452.
- [4] Morshed, A., Prakash Jayaraman, P., Sellis, T., Georgakopoulos, D., Villari, M. & Ranjan, R. (2017). Deep Osmosis: Holistic Distributed Deep Learning in Osmotic Computing in *Anais IEEE*. DOI: 10.1109/MCC.2018.1081070.
- [5] Li, P., Chen, Z., T. Yang, L., Zhang, Q. & Jamal Deen, M. (2017). Deep Convolutional Computation Model for Feature Learning on Big Data in Internet of Things. In *Anais do IEEE*. DOI: 10.1109/TII.2017.2739340.
- [6] Li, H., Ota, K. & Dong, M. (2018). Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. In *Anais do IEEE*. DOI: 10.1109/MNET.2018.1700202.
- [7] Zhu, J., Song, Y., Jiang, D. & Song, H. (2017). A New Deep-Q-Learning-Based Transmission Scheduling Mechanism for the Cognitive Internet of Things. In *Anais do IEEE*. DOI: 10.1109/JIOT.2017.2759728.
- [8] Zhang, H., Li, F., Wang, J., Wang, Z., Sanín, C. & Szczerbicki, E. (2017). Experience-Oriented Intelligence for Internet of Things. In *Anais do Cybernetics and Systems*. DOI: 10.1080/01969722.2016.1276771.
- [9] Yu, J., Kim, Y.-M., Kwon, S., Kim, K., Kim, N.-S., Kim, S.-J. & Pyo, C.-S. (2017). A Study on the Virtuous Circle Self-Learning Methods for Knowledge Enhancement. In *Anais da 2017 International Conference on Platform Technology and Service, PlatCon 2017 - Proceedings*. DOI: 10.1109/PlatCon.2017.7883683.
- [10] Li, J., Wang, X. & Su, H. (2016). Supermarket commodity identification using convolutional neural networks. In *Anais da Proceedings of 2016 2nd International Conference on Cloud Computing and Internet of Things, CCIOT 2016*. DOI: 10.1109/CCIOT.2016.7868315.
- [11] Valipour, S., Siam, M., Stroulia, E. & Jagersand, M. (2016). Parking-stall vacancy indicator system, based on deep convolutional neural networks. In *Anais do 2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016*. DOI: 10.1109/WF-IoT.2016.7845408.
- [12] Valipour, S., Siam, M., Stroulia, E. & Jagersand, M. (2017). Enabling Deep Learning on IoT Devices. In *Anais do IEEE*. DOI: 10.1109/MC.2017.3641648.
- [13] Ravi, D., Wong, C., Lo, B. & Yang, G.-Z. (2017). A Deep Learning Approach to on-Node Sensor Data Analytics for Mobile or Wearable Devices. In *Anais do IEEE*. DOI: 10.1109/JBHI.2016.2633287.
- [14] Liu, F., Shi, Y. & Li, P. (2017). Analysis of the Relation between Artificial Intelligence and the Internet from the Perspective of Brain Science. In *Anais da Procedia Computer Science*. DOI: 10.1016/j.procs.2017.11.383.
- [15] Park, S., Sohn, M., Jin, H. & Lee, H. (2016). Situation reasoning framework for the Internet of Things environments using deep learning results. In *Anais do IEEE*. DOI: 10.1109/ICKEA.2016.7803006.
- [16] Chen, F., Fu, Z. & Yang, Z. (2018). Wind power generation fault diagnosis based on deep learning model in internet of things (IoT) with clusters. In *Anais do Springer Science+Business Media, LLC, part of Springer Nature*. DOI: <https://doi.org/10.1007/s10586-018-2171-6>.
- [17] Kim, H. & Kim, J. (2016). A load balancing scheme based on deep-learning in IoT. In *Anais Springer Science+Business Media New York*. DOI: <https://doi.org/10.1007/s10586-016-0667-5>.
- [18] Ho Oh, S., Kim, G. & Lim, K. (2017). Compact deep learned feature-based face recognition for Visual Internet of Things. In *Anais Springer Science+Business Media, LLC, part of Springer Nature*. DOI: <https://doi.org/10.1007/s11227-017-2198-0>.
- [19] Liu, W., Clarence Yan, C., Liu, J. & Ma1, H. (2017). Deep learning based basketball video analysis for intelligent arena application. In *Anais Springer Science+Business Media, LLC*. DOI: <https://doi.org/10.1007/s11042-017-5002-5>.
- [20] Guo, J., Xie, X., Bie, R. & Sun, L. (2014). Structural health monitoring by using a sparse coding-based deep learning algorithm with wireless sensor networks. In *Anais Personal and Ubiquitous Computing*. DOI: <https://doi.org/10.1007/s00779-014-0800-5>.
- [21] Shi, X., Hao, Y., Zeng, D., Wang, L., Hossain, S., Md Mizanur Rahman, S. & Alelaiwi, A. (2016). Cloud-Assisted Mood Fatigue Detection System. In *Anais do Springer*. DOI: <https://doi.org/10.1007/s11036-016-0757-x>.
- [22] Zhang, D., Martinez, N., Lindholm, G. & Ratnaweera, H. (2018). Manage Sewer In-Line Storage Control Using Hydraulic Model and Recurrent Neural Network. In *Anais Springer Science+Business Media B.V., part of Springer Nature*. DOI: <https://doi.org/10.1007/s11269-018-1919-3>.
- [23] Chen, G., Wang, A., Zhao, S., Liu, L. & Chang, C. (2017). Latent feature learning for activity recognition using simple sensors in smart homes. In *Anais do Springer Science+Business Media, LLC*. DOI: <https://doi.org/10.1007/s11042-017-5100-4>.
- [24] Singh, H., Yadav, G., Mallaiah, R., Joshi, P., Joshi, V., Kaur, R., Bansal, S. & K. Brahmachari, S. (2017). iNICU – Integrated Neonatal Care Unit: Capturing Neonatal Journey in an Intelligent Data Way, In *Anais Journal of Medical Systems*. DOI: <https://doi.org/10.1007/s10916-017-0774-8>.
- [25] Wortmann, F. & Flüchter, K. (2015). Internet of Things: Technology and Value Added.
- [26] LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep Learning: Review.

ontoELICERE: Um Conjunto de Ontologias para Representar os Elementos de uma Análise de Perigos em Sistemas Aeroespaciais

Glauco da Silva¹, Carlos Henrique Netto Lahoz²

¹Instituto de Aeronáutica e Espaço

²Instituto Tecnológico de Aeronáutica

glaucogs@fab.mil.br, lahozchnl@fab.mil.br

Resumo: Este artigo apresenta o ontoELICERE, um conjunto de ontologias para representar os elementos de uma análise de perigos em sistemas aeroespaciais, suas relações, a técnica de análise utilizada, os perigos identificados e as medidas de mitigação sugeridas para cada perigo. Para se desenvolver o ontoELICERE, foi realizado um levantamento das informações necessárias para representar os elementos existentes na técnica ISTAR e quais informações são necessárias para a condução da análise de perigos. As ontologias foram criadas seguindo as recomendações do Método 101 e utilizou-se a linguagem OWL-DL. Com as ontologias é possível armazenar as medidas de mitigação selecionadas para que possam ser reutilizadas em outros projetos que venham a ser analisados, melhorando o processo de análise e o tempo de resposta. O ontoELICERE foi aplicado no acidente do Ariane 5 e foi útil na representação do modelo do sistema e na aplicação da análise de perigos. Todo conhecimento obtido com a análise é armazenado em bases de conhecimento para posterior consulta e reutilização em novas análises.

Palavras-chave: Ontologias, Análise de perigos, ELICERE.

1. Introdução

Atualmente, com o crescente número de aplicações críticas, como sistemas aeroespaciais, é necessário que a integridade do sistema como um todo seja garantida para que a missão não venha a falhar. A integridade pode ser relacionada a hardware, software, operação humana e ambiente. Neste sentido, a engenharia de segurança deve buscar o aprimoramento nos estudos e pesquisas de novas técnicas de análise de perigos, visto que poucas contribuições foram propostas tentando abranger esses quatro componentes e suas interações.

Nas investigações dos perigos são analisadas e descritas as potenciais situações em que o sistema pode ser levado a apresentar erros ou defeitos e conseqüentemente falhas. Estas análises buscam identificar comportamentos não planejados, troca de informações não previstas, entre outros problemas.

Várias técnicas são utilizadas para se realizar uma análise de perigos. Algumas técnicas bem conhecidas são FMEA (*Failure Mode and Effects Analysis*) [1], HAZOP (*Hazard and Operability Analysis*) [2] e FTA (*Fault Tree Analysis*) [3], que, apesar de serem consideradas tradicionais e conservadoras, ainda são muito utilizadas. Além destas técnicas, abordagens mais recentes, como STAMP (*Systems-Theoretic Accident Model and Processes*) [4], também tem sido adotadas. A maior dificuldade dessas abordagens é a dependência do conhecimento do analista, uma vez que é necessário que o condutor da análise conheça bem o sistema analisado.

Buscando minimizar estas dificuldades, em 2009 foi proposto o processo ELICERE [5]. Este processo integra a técnica de engenharia de requisitos orientados a metas ISTAR [6] com conceitos de técnicas de engenharia de segurança, como HAZOP e FMEA. O processo sugere como entrada um modelo do sistema a ser analisado e utiliza para análise um questionário que aborda características dos perigos potenciais de cada item do modelo. O resultado auxilia na identificação de metas de melhoria e medidas de mitigação, indicando atributos de qualidade ainda na fase de projeto.

O ELICERE foi proposto como um processo manual e gera um grande volume de informações. Além disso,

os resultados das análises e o conhecimento adquirido no preenchimento dos questionários não podem ser utilizados para análises de outros projetos, tornando-se necessária a criação de uma base de dados para armazenar este conhecimento. Buscando suprir essas necessidades, foi criada a ferramenta de automatização do ELICERE, o PRO-ELICERE [7, 8, 9, 10]. Nesta ferramenta foram desenvolvidas duas bases de dados, a de atributos (ADB) e a de conhecimento (KB).

Visando tornar a ferramenta PRO-ELICERE mais robusta e inteligente, foi desenvolvido um conjunto de ontologias, chamado ontoELICERE, para auxiliar a condução da análise de perigos por meio de inferências e descoberta de conhecimento nas bases de dados que armazenam técnicas e medidas de mitigação. Estas ontologias são utilizadas para representar todo o processo de análise do PRO-ELICERE, desde o modelo do sistema até as medidas de mitigação sugeridas.

2. Metodologia

No desenvolvimento do ontoELICERE, foi realizado um levantamento das informações necessárias para representar os elementos existentes na técnica ISTAR e para a condução da análise de perigos. Para se criarem as ontologias foi adotada a linguagem OWL-DL [11], em conjunto com o método 101 [12].

A aplicação do ontoELICERE no acidente do Ariane 5 é apresentada na Seção 4.

3. ontoELICERE

O ontoELICERE é composto por três ontologias: ontoISTAR, utilizada para representar os elementos do sistema modelados com a técnica ISTAR e analisar seus relacionamentos em termos de perigos e dependências; ontoQUESTION, utilizada para analisar os desvios de intenção de projeto sobre os campos do questionário; e ontoMITIGATION, aplicada para auxiliar na criação de medidas de mitigação. As ontologias foram criadas com o software livre Protégé¹ e são compostas por classes, atributos e relacionamentos. A Figura 1 apresenta as classes do ontoELICERE.

¹ <http://protege.stanford.edu/>

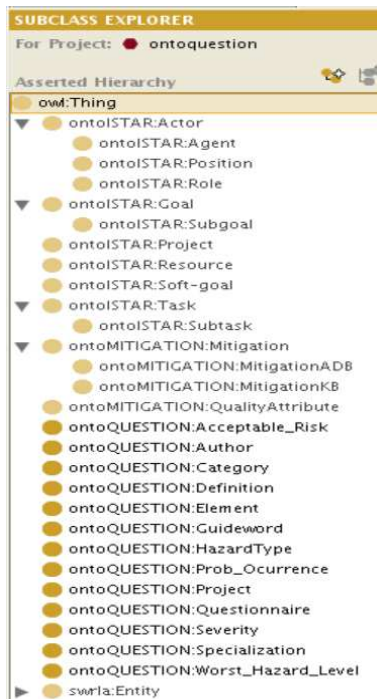


Figura 1: Classes do ontoELICERE.

3.1. ontoISTAR

A ontoISTAR foi criada baseada nos elementos existentes na técnica ISTAR [6], visando representar os elementos de um modelo ISTAR em instâncias da ontologia para que a análise de perigos seja conduzida.

Todas as classes definidas nesta ontologia são baseadas nos elementos que podem existir em um modelo ISTAR: ator, recurso, meta, meta-soft e outros. A ontologia também representa os relacionamentos entre os elementos, como níveis de dependência entre os elementos e decomposição de tarefas. As classes são utilizadas para conduzir a análise de perigo, permitindo definir a severidade, a palavra guia e assim por diante.

Uma das principais vantagens de se fazer uso de ontologias é a capacidade de inferir conhecimento para melhorar a condução da análise, incluindo, por exemplo, a possibilidade de detecção dos elementos que possuem maior número de dependências. Buscando identificar essa característica, foram definidas assertivas que apontam a quantidade de dependências de cada elemento. Exemplos dessas assertivas, representadas por descritores lógicos, são apresentados a seguir.

```
(ontoISTAR:Resource(?x) ^ ontoISTAR:dependencyFrom(?x,?y)) -> (sqwrl:select(?x) ^ sqwrl:count(?y))
(ontoISTAR:Resource(?x) ^ ontoISTAR:dependencyTo(?x,?y)) -> (sqwrl:select(?x) ^ sqwrl:count(?y))
```

Essas regras realizam a contagem de dependências para um elemento, por exemplo, se “?x” é um recurso (*ontoISTAR:Resource(?x)*) e ele possui dependência de um ou mais elementos (*ontoISTAR:dependencyFrom(?x,?y)*), ocorre a contagem deste número (*query:count(?y)*), na qual “?y” é o número de dependências de “?x”.

3.2. ontoQUESTION

As classes da ontoQUESTION foram criadas para representar os questionários e seus campos que devem

ser preenchidos durante a análise de perigos. Além de permitir a condução da análise, a ontoQUESTION também permite a descoberta de novas informações a respeito dos perigos analisados, como por exemplo, mecanismos para classificar o risco associado aos perigos analisados. Todos os riscos são associados com a possibilidade de ocorrência (*ontoQUESTION:Hazard_Probability*) e um nível de severidade (*ontoQUESTION:Hazard_Severity*). A seguir são apresentados alguns exemplos de como essa classificação pode ser utilizada.

Um perigo com ocorrência “*Improbable*” e severidade “*Catastrophic*” classifica o perigo com o risco “*Acceptable*”. Um perigo com ocorrência “*Frequent*” e severidade “*Major*” classifica o perigo com o risco “*Intolerable*”.

Com base nessas características, algumas assertivas foram definidas na ontologia para classificar o risco do perigo analisado. Essas assertivas, são representadas na ontologia por descritores lógicos, conforme apresentado no exemplo a seguir.

```
(ontoQUESTION:Element(?x) ^ ontoQUESTION:Hazard_Probability(?x, ontoQUESTION:Probability_Frequent) ^ ontoQUESTION:Hazard_Severity(?x, ontoQUESTION:Severity_Major)) -> (ontoQUESTION:Classification_of_Risk(?x, ontoQUESTION:Risk_Intolerable))
```

Esta regra define que se “?x” é um elemento analisado (*ontoQUESTION:Element*) e possui a probabilidade de ocorrência (*ontoQUESTION:Hazard_Probability()*) “*ontoQUESTION:Probability_Frequent*” e severidade (*ontoQUESTION:Hazard_Severity()*) “*ontoQUESTION:Severity_Major*”, então “?x” possui a classificação de risco (*ontoQUESTION:Classification_of_Risk()*) “*ontoQUESTION:Risk_Intolerable*”. Esta regra permite determinar o risco de cada perigo indicado na análise e então priorizar as análises dos perigos para aqueles com risco “*Intolerable*”.

Assim, quando um perigo apresentar um risco “*Intolerable*”, significa que o questionário associado a ele deve ser preenchido pelo analista/engenheiro, uma vez que este perigo pode causar um comprometimento do sistema, sendo assim é necessário aplicar alguma medida de mitigação para reduzir ou eliminar o risco do sistema deixar de funcionar.

3.3. ontoMITIGATION

Para se identificarem as medidas de mitigação, o sistema pode utilizar as bases de dados de atributos (ADB) e a base de conhecimentos (KB). A ADB foi criada utilizando-se trabalhos da literatura [13, 14, 15, 16], que conduziram levantamentos de perigos, suas causas e suas possíveis mitigações. A KB foi construída por meio de medidas de mitigação utilizadas em projetos já analisados, permitindo o reuso de mitigações já adotadas. Todas as medidas de mitigação existentes são representadas por instâncias desta ontologia.

Para popular a KB por meio da ontologia, foi criada uma assertiva que define que qualquer medida de mitigação adotada para um questionário deve ser adicionada na KB, incluindo as informações do projeto em que ela foi aplicada e para qual perigo foi selecionada. Essa assertiva é representada na ontologia por descritores lógicos, conforme apresentado a seguir.

$(\text{ontoQUESTION:Questionnaire}(?x) \wedge \text{ontoQUESTION:questionnaireMitigation}(?x, ?y) \wedge \text{ontoQUESTION:OriginProject}(?x, ?z)) \rightarrow (\text{ontoMITIGATION:MitigationKB}(?y) \wedge \text{ontoMITIGATION:Project}(?y, ?z))$

Essa assertiva indica que se existe um questionário “?x” ($\text{ontoQUESTION:Questionnaire}(?x)$) e “?y” é uma medida de mitigação adotada para “?x” ($\text{ontoQUESTION:questionnaireMitigation}(?x, ?y)$), e “?z” é o projeto original do questionário “?x” ($\text{ontoQUESTION:OriginProject}(?x, ?z)$), então “?y” é uma medida de mitigação que deve ser inserida na base de conhecimento ($\text{ontoMITIGATION:MitigationKB}(?y)$) e a informação de qual projeto utilizou aquela mitigação deve ser inserida acompanhada de a medida adotada ($\text{ontoMITIGATION:Project}(?y, ?z)$).

Quando o sistema busca por uma medida de mitigação para um perigo, primeiramente é realizada uma busca na KB, visando fazer o reuso de uma mitigação já adotada para um perigo similar de outro projeto. Caso o sistema não encontre a mitigação na KB, então uma busca é realizada na ADB.

4. Uma aplicação do ontoELICERE

A família de veículos lançadores de satélites Ariane do *Centre National D’etudes Spatiales* (CNES), tem como objetivo colocar satélites artificiais em órbitas geoestacionárias, e também enviar cargas para órbitas de baixa altitude.

Em 4 de junho de 1996, o primeiro voo do lançador Ariane 5 não foi bem sucedido. Aproximadamente 40 segundos após o início da sequência de voo, a uma altitude de cerca de 2.7 Km, o lançador se desviou de sua trajetória, se quebrou e explodiu [18]. O relatório de acidente [17] descreve a “causa primária” como a perda total das informações de orientação e atitude 37 segundos após o início da sequência de ignição do motor principal (30 segundos após a decolagem). A perda de informação ocorreu devido a erros de especificação e de projeto no software do sistema de referência inercial [18]. O software foi reutilizado a partir do Ariane 4 e incluiu funções que não eram necessárias para o Ariane 5. As perdas do acidente incluíram cinco bilhões de dólares de carga útil [17].

Após estudos sobre a estrutura do Ariane 5 e a identificação de seus elementos, foi desenvolvido o modelo ISTAR para o veículo lançador e identificadas as metas, metas-soft, recursos e tarefas, assim como as dependências existentes entre eles.

A partir do modelo, a ontoISTAR foi utilizada para representar os elementos existentes, tornando possível a condução da análise de perigos. As instâncias representam os elementos e suas características, bem como suas dependências com outros itens do modelo.

Após a representação do modelo pela ontoISTAR, a ontoQUESTION foi utilizada para representar a Tarefa “*Calculate inertial data*”, que foi identificada como um dos elementos mais críticos do sistema. Vale ressaltar que a ontologia representa todos os elementos do sistema, uma vez que outros perigos podem existir.

Após o preenchimento do questionário, são apresentadas as sugestões de mitigação, representadas por instâncias da ontoMITIGATION. Primeiro as da

KB, e caso não exista nenhuma ou a apresentada não seja satisfatória, são apresentadas as da ADB. O objetivo é identificar se na ADB existem técnicas que possam produzir recomendações adequadas ao problema. Após a seleção de uma ou mais medidas, os resultados são inseridos na KB como mitigações já utilizadas, assim é possível que projetos futuros possam fazer o reuso desta informação.

As recomendações indicadas no relatório de investigação [17] também foram inseridas na KB, pois são conhecimentos já reportados e portanto podem ser utilizados em projetos futuros semelhantes.

Para o caso analisado, foi verificado que o cálculo incorreto ocorreu devido ao reuso de uma parte do software do Ariane 4. Para este caso não foi encontrada nenhuma medida de mitigação na KB, portanto a busca foi realizada na ADB, e teve como sugestão a medida de mitigação “*Reuse checklist*” (Figura 2), indicando recomendações que devem ser seguidas para garantir o reuso de código de maneira eficiente. Alguns trabalhos abordam o reuso de software [19, 20], mas basicamente as recomendações envolvem testes intensivos nos códigos reutilizados e risco aceitável desses códigos

ontoMITIGATION:Mitigation_Task_Software...	
ontoMITIGATION:Project =	ontoQUESTION:Project_Ariane_5
ontoMITIGATION:QualityAttributeADB =	ontoMITIGATION:QualityAttribute_Reusabil...
ontoMITIGATION:MitigationHazard =	ontoQUESTION:Hazard_Task_Incorrect
ontoMITIGATION:MitigationHazardType =	ontoQUESTION:HazardType_Software
ontoQUESTION:MitigationDescription =	Reuse Checklist

Figura 2: Medida de mitigação “Reuse checklist” representada pelo ontoMITIGATION.

5. Conclusões

A necessidade de se realizarem análises de perigos em sistemas críticos tem motivado o desenvolvimento de novas ferramentas para auxiliar nesta tarefa. Os desafios de pesquisa englobam técnicas, geralmente utilizadas em dispositivos mecânicos ou eletrônicos, que precisam ser adaptadas para software.

Além deste desafio, também é necessária a criação de ferramentas que auxiliem na condução da análise. Devido a essa necessidade, pensou-se na criação do PRO-ELICERE, que busca automatizar o processo ELICERE para análise de perigos.

A versão inicial do PRO-ELICERE foi utilizada para introduzir a aplicação do ontoELICERE, mostrando como ele deve operar com esta nova funcionalidade. O preenchimento do questionário de forma eletrônica, com valores pré-fixados, não gera ambiguidade nem dúvidas quanto aos elementos que devem ser inseridos.

O processo se mostrou robusto para realizar as recomendações de mitigação, com base nas técnicas de mitigação existentes na ADB e KB. Isto mostra que o uso das ontologias é capaz de auxiliar na condução da análise. A utilização do ontoELICERE se apresenta como uma ferramenta poderosa para suporte a decisão, uma vez que permite a adoção ou não da medida sugerida para o perigo.

Além disso, o ontoELICERE se mostrou promissor na representação dos elementos existentes no sistema analisado. As ontologias são úteis para a realização de inferências, permitindo que novas informações e novos conhecimentos sobre o sistema analisado sejam obtidas.

Bibliografia

- [1] Stamatis, D. H. (2003) Failure mode and effect analysis: FMEA from theory to execution. ASQ Quality Press.
- [2] Redmill, F. et al. (1999) System safety: HAZOP and software HAZOP. Wiley Chichester.
- [3] Lee, W.-S. et al. (1985). Fault tree analysis, methods, and applications: A review. *IEEE transactions on reliability* 34(3):194-203. DOI: 10.1109/TR.1985.5222114
- [4] Leveson, N. (2004) A new accident model for engineering safer systems. *Safety science*, 42(4):237-270. DOI: 10.1016/S0925-7535(03)00047-X
- [5] Lahoz, C. H. N. (2009) ELICERE: O processo de elicitação de metas de dependabilidade para sistemas computacionais críticos: Estudo de caso aplicado à área espacial. Tese de Doutorado. Universidade de São Paulo. Programa de Pós-Graduação em Engenharia Elétrica.
- [6] Yu, E. S. (1995) Modeling Strategic Relationship for Process Reengineering. Tese de Doutorado. Toronto University.
- [7] Silva, G.; Lahoz, C. H. N. (2013) A new process for space computer system dependability analysis. In *64th International Astronautical Congress*, Beijing, China. IAF.
- [8] Silva, G.; Lahoz, C. H. N. (2013) Pro-elicere: A study for create a new process for dependability analysis of space computer systems. In *Sixth IAASS Conference - Safety is Not an Option*, Montreal, Canada. ESA Communications.
- [9] Pivetta, T. A. (2016) Uma estrutura de base de dados para o processo de análise de perigos - pro-elicere. Dissertação de Mestrado. Instituto Tecnológico de Aeronáutica. Programa de Pós-Graduação em Ciências e Tecnologias Espaciais.
- [10] Pivetta, T. A. et al. (2016) Pro-elicere: A hazard analysis automation process applied to space systems. *Journal of Aerospace Technology and Management* 8(3):328-338. DOI: 10.5028/jatm.v8i3.609
- [11] Horrocks, I. et al. (2003) From shiq and rdf to owl: the making of a web ontology language. *Web Semantics: Science, Services and Agents on the World Wide Web* 1(1):7-26. DOI: 10.1016/j.websem.2003.07.001
- [12] Noy, N. F.; McGuinness, D. L. (2001) Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Knowledge Systems, AI Laboratory, Stanford University, Stanford, EUA.
- [13] Avizienis, A. et al. (2004) Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing* 1(1):11-33. DOI: 10.1109/TDSC.2004.2
- [14] Bass, L. (2007). Software architecture in practice. Pearson Education India.
- [15] Romani, M. A. S. et al. (2010). Identifying dependability requirements for space software systems. *Journal of Aerospace Technology and Management* 2(3):287-300. DOI: 10.5028/jatm.2010.02037810
- [16] Ericson, C. A. et al. (2015) Hazard analysis techniques for system safety. John Wiley & Sons.
- [17] Lions, J.-L. et al. (1996) Ariane 5 flight 501 failure. Technical report, European Space Agency.
- [18] Leveson, N. G. (2004) Role of software in spacecraft accidents. *Journal of Spacecraft and Rockets* 41(4):564-575. DOI: 10.2514/1.11950
- [19] Leach, R. J. (1997) Software Reuse: methods, models, and costs. McGraw-Hill New York.
- [20] Rothenberger, M. A. et al. (2003) Strategies for software reuse: A principal component analysis of reuse practices. *IEEE Transactions on Software Engineering* 29(9):825-837. DOI: 10.1109/TSE.2003.1232287

Redesign do Serious Game Caixa de Pandora para Dispositivos Móveis

Júlio R. Oliveira^{1,3}, Zildomar C. Félix³, Luana R. Almeida², Ana Tereza Medeiros³, Liliane S. Machado^{1,3}

¹Departamento de Informática, ²Departamento de Enfermagem em Saúde Coletiva,

³Laboratório de Tecnologias para o Ensino Virtual e Estatística (LabTEVE)

Universidade Federal da Paraíba

julio.rapha@hotmail.com, zildomarf@gmail.com, luanaralmeida02@gmail.com, anaterzaprof@gmail.com, liliane@di.ufpb.br

Palavras-chave: *Serious Games, Caixa de Pandora, Android Apps.*

Introdução. *Serious Games* são jogos que possuem um propósito específico que extrapola a ideia de entreter para oferecer novos tipos de experiência aos seus usuários [1]. Caixa de Pandora é um *serious game* desenvolvido por Almeida *et al.* [2] para execução em computadores pessoais, que possui como objetivo a capacitação dos profissionais da área da saúde sobre a violência contra a mulher, por meios de passatempos que foram construídos a partir dos subtemas Gênero, Direitos Humanos e Saúde. A partir da observação da necessidade de jogos sobre esta temática em plataformas móveis, bem como a necessidade de disseminação do tema para o público geral, foi proposto o processo de *redesign* do jogo. Assim, o objetivo deste trabalho é descrever como ocorreu este processo de *redesign*, apresentando resultados de uma avaliação preliminar realizada. **Metodologia.** O processo de *redesign* ocorreu em três etapas: construção, adequação da base de conhecimento e testes. Para a primeira etapa foi utilizada a *game engine* Defold junto com a linguagem de programação LUA para a programação do jogo e sua inteligência. Nessa fase foi possível criar um sistema dinâmico de apresentação das alternativas de resposta, além da utilização de pacotes de edição de imagens, texto e áudio. Na segunda etapa, visando ampliar o público alvo do jogo, toda a base conhecimento passou por um processo de revisão do conteúdo, de modo que a terminologia e a forma de abordagem dos conceitos foram adequadas. Finalmente, na terceira etapa, o jogo foi disponibilizado para testes em um grupo restrito de 30 alunos de graduação e pós-graduação de duas instituições de ensino superior. Os alunos foram instruídos a instalar o jogo em seus dispositivos móveis, jogá-lo e, posteriormente responder a um questionário sobre satisfação [3][4]. **Resultados.** O novo jogo (Figura 1) foi disponibilizado na plataforma GooglePlay e, a partir da análise das respostas dos questionários, foi possível verificar que os usuários sentiram-se satisfeitos em todas as dimensões avaliadas. Porém, dentre os resultados obtidos, a dimensão da “Imersão” chamou a atenção por evidenciar a satisfação apenas com a imersão emocional, e não com a imersão física. **Discussão.** Os resultados obtidos permitiram observar que os objetivos foram alcançados, bem como despertar a possibilidade de utilização de novas tecnologias que permitam alcançar a imersão física dos usuários. O jogo está disponível em <https://play.google.com/store/apps/details?id=br.ufpb.labteve.caixapandora>.

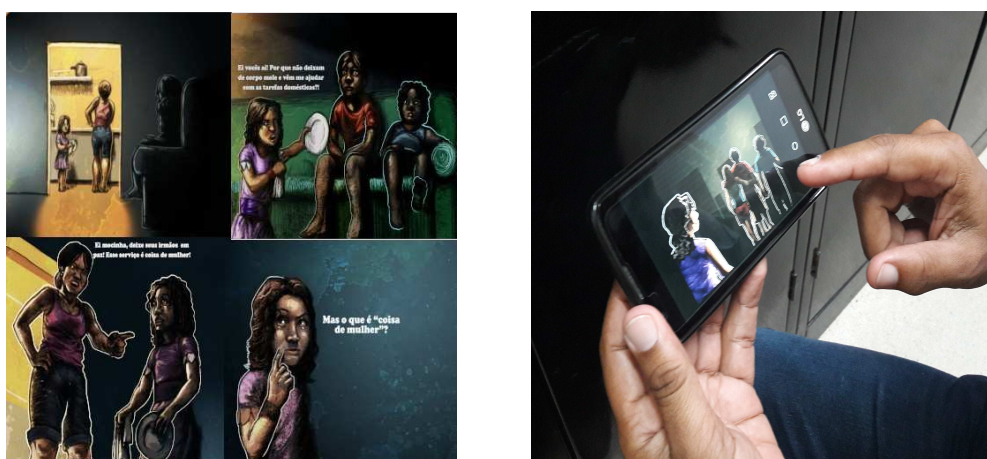


Figura 1. Imagem dos passatempos e execução no dispositivo móvel.

Bibliografia

- [1] Machado, L. S. et al. (2011) Serious games baseados em realidade virtual para educação médica. Rev. Bras. Educ. Med. 35(2): 254-262. DOI: 10.1590/S0100-55022011000200015
- [2] Almeida, L.R.; Machado, L.S.; Medeiros, A.T. et al. (2018) The Caixa de Pandora game: changing behaviors and attitudes towards violence against women. ACM Computers in Entertainment 16(3). DOI: 10.1145/3236493
- [3] Sweetser, P.; Wyeth, P. (2005) GameFlow: a model for evaluating player enjoyment in games. Computers in Entertainment (CIE) 3(3). DOI: 10.1145/1077246.1077253
- [4] Rigby, S.; Ryan, R. (2007) The player experience of need satisfaction (PENS) model. Immersyve Inc, 22p. Online: <http://immersyve.com/white-paper-the-player-experience-of-need-satisfaction-pens-2007/#>

Dungeon Scrolls - Geração de Conteúdo para Jogos RPG

Armando S. Salvador Neto, Drayton C. Filho, Douglas F. L. Lima, Ewerton N. Santos

Centro de Informática

Universidade Federal da Paraíba

armando_tgt@hotmail.com, drayton80@hotmail.com, douglasliralima@gmail.com, EwertonDNSantos@gmail.com

Palavras-chave: RPG, Machine Learning, Aplicação Android

Introdução. Dungeon Scrolls é um aplicativo que visa auxiliar jogadores de *role-playing game* (RPG), mais especificamente RPG de mesa, utilizando de um sistema de salas, armazenamento de textos, cálculos automatizados e aplicando inteligência artificial para geração de criaturas. **Motivação.** Ao pensar em jogos de RPG de mesa no passado, quanto mais autoral uma história ficava, mais complexa tornava-se a campanha, devido ao volume de regras e características de qualquer sistema. Outro fator de dificuldade vinha das informações da campanha que eram registradas por meio de papel e caneta o que levava iniciantes a procurar histórias prontas com possibilidades limitadas, mas que facilitassem o jogo. Hoje devido a ferramentas como o software brasileiro Firecast ou o conhecido Roll20, partidas de RPG vem recebendo um novo público, devido a facilidades, como fichas que são organizadas em um banco de dados na nuvem, cálculos que são feitos automaticamente, abstração de regras e outras facilidades. Nosso intuito é fabricar uma ferramenta que se assemelha a essas gerenciadores de partidas, mas dando um passo maior na facilidade criativa dada ao jogador, usando de técnicas de inteligência artificial para trazer a possibilidade da criação de monstros. **Metodologia.** No lado da aplicação Android, preferiu-se usar uma arquitetura de desenvolvimento de software, o Model-View-ViewModel (MVVM) com Databinding [1], trazendo maior padronização de código e um sistema de inteligência artificial usando um servidor Python, para o uso de suas bibliotecas na aplicação dos métodos de machine learning. Na falta de uma base de dados sobre monstros em qualquer sistema de RPG, foi coletado uma do sistema Dungeons & Dragons [2], devido a vasta literatura e popularidade do sistema. Usou-se a técnica de aprendizagem não supervisionada Mean Shift [3] para melhor visualização da base, observando suas necessidades de balanceamento e correlação dos atributos, de modo a melhorar nossas previsões. Na determinação de monstros artificiais, foi usada uma árvore de decisão regressiva [3] usando atributos baseados na necessidade do usuário e na correlação desses com o dos demais monstros na base de dados. Na persistência de dados foi usado o Firestore do Firebase para compartilhar as informações da aplicação com todos os usuários. Como era necessário que fosse possível visualizar os dados mesmo sem internet, fez-se uso do SQLite de modo que os dados baixados do Firestore ficassem salvos no aparelho, deixando, assim, o aplicativo mais flexível. **Resultado.** Foi criado um aplicativo que auxilia na criação de fichas de RPG. O aplicativo conta com um sistema baseado em salas e o mestre de cada sala tem a possibilidade de criar fichas de monstros, sem misturá-las com as de outros jogadores. A criação das fichas de monstros pode se dar de forma manual ou através da geração automática, feita com Machine Learning e utilizando a base de dados que criamos do sistema de Dungeons & Dragons cujas instâncias foram coletadas com base nos livros oficiais [2]. **Discussão.** A base de dados que foi criada para o desenvolvimento do projeto levou em conta apenas o primeiro dos cinco volumes do Monster Manual, trazendo uma pequena amostragem de elementos, logo, a cobertura de mais instâncias para o treinamento da Inteligência artificial seria uma melhoria na criação dos monstros gerados, principalmente aqueles categorizados com o nível de dificuldade elevado os quais haviam poucas instâncias, levando a resultados não tão precisos quando o objetivo era gerar um dessa categoria. Além disso, uma possível adição futura tendo em vista a aplicação Android, seria a adesão de um mapa dando senso de localização aos jogadores. Movendo-se para o âmbito de contribuições, é tido que as mais relevantes obtidas ao longo do projeto foram a criação da base de dados, a qual pode ser obtida via requisição por *email* para alguns de seus criadores, e o levantamento de atributos mais importantes para o uso da mesma.

Bibliografia

[1] Guide to App Architecture. Online: <https://developer.android.com/jetpack/docs/guide>. Acesso em 15/04/2018.

[2] Williams, S. (2003) Dungeons and Dragons Monster Manual, 3.5ª edição. Wizards of the Coast.

[3] Faceli, K. (2015) Inteligência Artificial: Uma abordagem de aprendizado de máquina, 1ª edição, LTC - Livros Técnicos e Científicos. Editora Ltda.

Reutilização de Lâmpadas Fluorescentes Econômicas Descartadas

Thales M. Soares^{1,2}, Gustavo E. Silva^{1,2}, Leandro M. Santos^{1,2}, Mardson F. Amorim^{1,2,3}, Antonio J. G. Oliveira¹, José G. Batista Neto¹, Jordy A.S. Lima^{1,2}, Diógenes F.X. Andrade^{1,2}, Antonio C. Cavalcanti^{1,2,3}, Alexandre R. Lopes Filho¹

¹Centro de Informática, ²Laboratório de Medição e Instrumentação, ³Departamento de Sistema de Computação
Universidade Federal da Paraíba

{thales.monteiro;gustavoeraldo;leandro.mendes}@eng.ci.ufpb.br, mdsamorim@gmail.com,
{antoniojonas11;josegabrielpe}@hotmail.com, jordy.sousa@eng.ci.ufpb.br, diogenesfernando363@outlook.com,
caval@ci.ufpb.br, alexandrerdriqueslopesfilho@gmail.com

Palavras-chave: reciclagem, lâmpadas fluorescentes, mercúrio.

Introdução. As lâmpadas fluorescentes ganharam bastante espaço no mercado nos últimos anos, porém é uma tecnologia que está com os dias contados. Tais lâmpadas contêm vapor de mercúrio, um metal pesado e perigoso ao meio ambiente, pois é um poluente tóxico, persistente e bioacumulativo, que está se dispersando no meio ambiente através do descarte inadequado das lâmpadas [1]. Este projeto visa conscientizar sobre o desperdício dos materiais das lâmpadas fluorescentes juntamente com os riscos gerados ao meio ambiente e aos seres vivos. Além disso, tem-se o foco em ensinar a catadores como deve-se manusear as lâmpadas para fazer a reciclagem/modificação do material eletrônico já que em cerca de 80% das lâmpadas, se encontra com sua parte eletrônica reaproveitável [2]. **Metodologia.** Foi realizada uma visita técnica na Universidade Federal da Paraíba e observamos os métodos de descarte do material, onde o *Bulb-eater* (papa lâmpada) é o meio reciclagem mais utilizado. Em seguida utilizamos o conceito de engenharia reversa para compreender o funcionamento do sistema eletrônico das lâmpadas e fazer as alterações necessárias no circuito adaptando-o a um funcionamento de uma lâmpada de LED. **Resultados.** Após o processo de adaptação e modificação do circuito eletrônico (Figura 1) obteve-se o resultado esperado e o protótipo funcional está operando em condições ideais. **Discussão.** No Brasil, um projeto com iniciativa de reaproveitar as lâmpadas fluorescentes após a descontaminação do material químico mercúrio, foi criado pela empresa Radak Sul [3] que tinha como principal objetivo complementar o trabalho de reciclagem, reutilizando componente do produto que ainda é funcional, e promovendo uma nova aplicação para ele, gerando uma nova forma de rentabilidade. **Conclusão.** Encontramos uma maneira viável da reutilização dos componentes eletrônicos em prol do funcionamento sem dependência do mercúrio. Além disso, vale destacar a importância da conscientização trabalhada e desenvolvida pelo grupo em relação aos perigos oferecidos pelo mercúrio e outros componentes presentes nas lâmpadas.

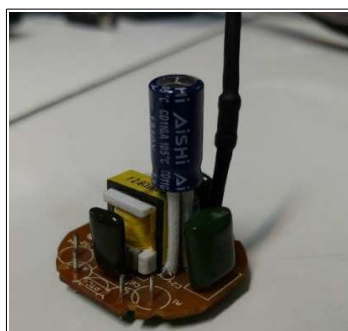


Figura 1. Circuito Eletrônico.

Bibliografia

- [1] Mombach, V. L.; Riella, H. G.; Kuhnen, N. C. (2008) O Estado da Arte na Reciclagem de Lâmpadas Fluorescentes no Brasil: Parte 1. Acta Ambiental Catarinense 5(1/2). DOI: <http://dx.doi.org/10.24021/raac.v5i1/2.225>.
[2] eCycle. Online: <https://www.ecycle.com.br/591-lampada-fluorescente>. Acesso em 12/04/2018.
[3] Radak Sul. Online: <http://www.radak.com.br/index.php?> Acesso em 12/04/2018.