

Integrando BPMN e AIML para construção de fluxos de diálogo para Chatbots

Giseldo S. Neo², Evandro C. Barros¹, Alana V. B. S. Neo²,
J. Antão B. Moura², Arthur C. Correia³, Wanderson R. M. Rodrigues¹

¹Universidade Federal de Alagoas (UFAL) - Maceió - Alagoas - AL - Brasil

²Universidade Federal de Campina Grande (UFCG) - Campina Grande - PB - Brasil

³Universidade Federal de Pernambuco (UFPE) - Recife - PE - Brasil

{giseldo, alana.viana}@copin.ufcg.edu.br, antao@computacao.ufcg.edu.br

cacc3@cin.ufpe.br, ebc.academico@ic.ufal.br, rubian64@gmail.com

Abstract. *Among the languages for specifying the knowledge base of chatbots, there is one that stands out for its simplicity: the Artificial Intelligence Markup Language (AIML). However, designing dialog flows with AIML requires knowledge of their XML tags; in addition, the more complex the flow, the greater the effort to maintain this structure. On the other hand, the Business Process Model and Notation (BPMN) specification provides a series of symbols and patterns for easy visual interpretation. To minimize the complexity in building and maintaining AIML XML flows, we propose a BPMN to AIML converter (BPMN2AIML). To validate the proposal, we carried out a case study where users mapped dialog flows and evaluated their usability, we also created two chatbots and evaluated the interaction of one of them. The experimental results demonstrate that designing visual flows to represent the knowledge base of AIML chatbots with BPMN is possible and apparently more intuitive than working directly with XML.*

Resumo. *Entre as linguagens para a especificação da base de conhecimento de chatbots existe uma que se destaca pela sua simplicidade o Artificial Intelligence Markup Language (AIML). Porém, desenhar fluxos de diálogo com AIML exige o conhecimento de suas tags XML, além disso, quanto mais complexo o fluxo maior o esforço de manutenção dessa estrutura. Por outro lado, a especificação Business Process Model and Notation (BPMN) prevê uma série de símbolos e padrões de fácil interpretação visual. Para minimizar a complexidade na construção e manutenção dos fluxos em XML do AIML propomos um conversor do BPMN para o AIML (BPMN2AIML). Para Validar a proposta realizamos um estudo de caso em que usuários mapearam fluxos de diálogo e avaliaram a sua usabilidade, também criamos dois chatbots e avaliamos a interação de um deles. Os resultados experimentais demonstram que desenhar fluxos visuais para representar a base de conhecimento dos chatbots AIML com BPMN é possível e aparentemente mais intuitivo do que trabalhar direto com o XML.*

1. Introdução

Um chatbot é um programa que interage com o ser humano utilizando linguagem natural, sendo projetado para oferecer respostas diretas e buscando manter relativa coerência

[Barros and Tedesco 2016]. Um marco inicial na área de processamento de linguagem natural (PLN) é o seminal artigo de Turing [Turing 1950]. Neste artigo ele descreve um teste para verificar se uma máquina pode pensar. Um dos primeiros chatbots que se destacou neste teste foi a “Eliza” [Weizenbaum 1966]. Ela utilizava técnicas simples de casamento de padrão, mas surpreendeu a comunidade não científica com a sua capacidade em manter usuários conversando com o programa por longos períodos [Shum et al. 2018].

Várias técnicas de engenharia de software foram propostas desde “Eliza” para uso em agentes conversacionais. Uma evolução das técnicas de casamento do Padrão utilizadas na “Eliza” é o *Artificial Intelligence Markup Language (AIML)*. Essa linguagem vem há décadas sendo utilizada na construção de agentes conversacionais e provê um ecossistema padronizado, robusto e confiável [Wallace 2009]. Mudanças na linguagem AIML são organizadas pela *AIML Foundation*, uma fundação sem fins lucrativos que disponibiliza e padroniza a linguagem. Um destaque dessa abordagem é o uso da teoria estímulo-resposta e a representação da base de conhecimento do chatbot em XML, permitindo diálogos mais complexos organizados em fluxos.

O *chatGPT* utiliza outra técnica, um modelo multimodal de grande escala. Porém, em situações em que a confiabilidade é importante, chatbots desse tipo não são recomendados [OpenAI 2023]. Já os AIML chatbots funcionam bem em domínios específicos, pois são adequados em situações em que existam intenções bem definidas, por exemplo, abrir uma conta em um banco ou agendar uma passagem aérea, dada a sua estrutura controlada e sua saída previsível. O que não acontece com chatbots semelhantes ao *chatGPT* que, conforme [OpenAI 2023], não são totalmente confiáveis, podem sofrer de “alucinações”, têm uma janela de contexto limitada e não aprendem. Diferente dos chatbots AIML que podem aprender com a interação e se aproveitam da existência de ontologias.

Um fluxo de diálogo é um conjunto de perguntas e respostas envolvidos em um contexto. Construir fluxos de diálogos com AIML envolve a utilização de diversas *tags* em XML. Quanto maior o fluxo maior a chance de confusão e erros na manutenção desse XML. Por exemplo, o chatbot “ALICE” é um AIML chatbot com 16.000 categorias, onde cada categoria é um conjunto de *tags* XML que podem conter várias outras *tags* aninhadas [Wallace 2009].

A construção desses fluxos é desafiadora para os engenheiros de software, muitas vezes é difícil ver como as categorias se vinculam, por ser um formato baseado em texto [AISB 2019]. A plataforma comercial AIML *Pandorabots* [Larisane Kuyven et al. 2018] criou uma solução visual para superar essa dificuldade, mas até o momento da realização deste estudo ela não tinha uma boa representação para os fluxos do diálogo. Quando comparamos AIML chatbots com alguns chatbots não AIML (*Watson*, *Dialog Flow* e *Rasa*) notamos que seus concorrentes oferecem diversas formas de desenhar fluxos.

Existem algumas ferramentas para facilitar o uso do AIML que propõem abordagens visuais [Oladele et al. 2017, Silveira 2018]. Porém, a maioria constrói suas próprias interfaces proprietárias e dispõem de recursos visuais limitados. Um padrão aberto, robusto, replicável, aproveitando grandes ecossistemas de bibliotecas e ferramentas, no melhor dos nossos esforços, não foi encontrado.

Por outro lado, o *Business Process Model and Notation (BPMN)* é uma especificação utilizada na construção de Processos de Negócio que internamente utiliza XML.

Ela estabelece uma série de símbolos visuais e vem sendo usada extensivamente por profissionais de negócios para modelar processos [ABPMP 2013].

Dado esse contexto, a hipótese é que os recursos visuais do BPMN na construção dos fluxos de diálogo AIML facilitam a criação da base de conhecimento dos AIML chatbots. Assim, chatbots que interpretam AIML poderiam ter seus fluxos desenhados em várias das ferramentas visuais que já suportam o padrão BPMN. Desse modo evita-se a necessidade de criar toda uma aplicação visual, pois o BPMN já tem diversas ferramentas visuais para modelagem, reutilizando todo o ecossistema disponível do BPMN na construção dos AIML chatbots.

O objetivo deste estudo é diminuir o problema da complexidade da manutenção e criação de fluxos de diálogo AIML, para isso propomos um algoritmo de conversão de fluxos desenhado em BPMN para o padrão AIML, chamado BPMN2AIML. Permitindo que bases de conhecimento dos AIML chatbots possam ser modeladas em BPMN.

Para validar essa proposta projetamos um algoritmo, implementamos uma prova de conceito e fizemos experimentos com usuários. Alguns usuários foram treinados na proposta e criaram vários fluxos com o uso desta notação, enquanto outros conversaram com os chatbots construídos de forma controlada. A prova de conceito, com esse conversor, com o chatbot e com documentação está disponível no *github*¹.

A originalidade desta pesquisa reside no fato da possibilidade de representar toda a base de conhecimento de um AIML chatbot em diagramas visuais BPMN. Além disso, o algoritmo elaborado para esta transformação pode ser reutilizado ou estendido para suportar outros tipos de componentes visuais.

2. Fundamentação Teórica

2.1. AIML

O *Artificial Intelligence Markup Language* (AIML) é uma especificação em XML para a programação de chatbots. O foco do design dessa especificação é o minimalismo. A simplicidade do AIML permite que qualquer pessoa que conheça um pouco de HTML possa escrever bases de conhecimento para seus chatbots [Wallace 2009]. Um exemplo de uma estrutura de um arquivo AIML pode ser visto na Figura 1.

```
<aiml> <category>
<pattern>OLÁ *</pattern>
<template>COMO VOCÊ ESTÁ</template>
</category> ... </aiml>
```

Figura 1. Exemplo de um arquivo AIML.

A teoria “Case-Based Reasoning” (CBR) mapeia bem o algoritmo de um interpretador AIML. Os casos “CBR” são as categorias no AIML. O algoritmo encontra o padrão de melhor correspondência para cada entrada. A categoria vincula o modelo de resposta diretamente ao padrão de estímulo. Um chatbot que utiliza AIML também pode ser classificado como pertencente a tradição da robótica “minimalista”, “reativa” ou “estímulo-resposta” [Wallace 2009].

¹https://github.com/giseldo/chatbot_BTA_BPMN_to_AIML.

Existem diversas classificações propostas para chatbots. Uma classificação que se adapta bem aos chatbots em AIML é a denominação “baseados em recuperação”. A principal característica dessa abordagem é utilizar um repositório de respostas predefinidas e selecionar uma resposta apropriada com base na entrada e no contexto. Neste tipo de técnica os usuários que conversam com o chatbot são guiados por um fluxo de interação bem definido, onde, foi anteriormente definida uma base de conhecimento. Geralmente esses chatbots lidam em um domínio restrito e com objetivos bem definidos [Borah et al. 2019].

Um chatbot AIML tem sua base de conhecimento representada na linguagem AIML. O procedimento para configurar um chatbot desse tipo é pegar esse arquivo AIML válido, criado em qualquer editor de texto e copiar para um diretório específico. Existem vários interpretadores de chatbots compatíveis com AIML que podem ser utilizados. Esse ecossistema ao redor do AIML facilita e muito a construção de chatbots para diversos fins.

2.2. BPMN

O *Business Process Model and Notation* (BPMN) é um padrão criado pela *Business Process Management Initiative* (BPMI) e incorporado ao grupo que estabelece padrões para sistemas de informação, o *Object Management Group* (OMG). A aceitação do BPMN cresce com o suporte que várias ferramentas de modelagem vêm oferecendo a essa especificação. Essa notação apresenta um conjunto robusto de símbolos para modelagem de diferentes aspectos de processos de negócio. Como na maioria das notações, os símbolos descrevem relacionamentos claramente definidos, tais como fluxo de atividades e ordem de precedência [ABPMP 2013].

Um processo de negócio, modelado por essa notação, abrange vários eventos e atividades. Eventos correspondem a coisas que acontecem atômicamente, o que significa que não têm duração. Um evento pode desencadear a execução de uma série de atividades. Além de eventos e atividades, um processo típico envolve pontos de decisão, ou seja, momentos em que uma decisão afeta a maneira como o processo é executado [Dumas et al. 2013].

Um processo de negócio já mapeado também pode ser representado em notação XML. A representação textual de um processo mapeado utilizando notação BPMN pode ser verificada Figura 2.

```
<bmn> <process id="0">
<startEvent id="1" name="name1" \>
<task id="3" name="name2" \>
<sequenceFlow id="2" sourceRef="1" targetRef="3" \>
<exclusiveGateway id="6">
<incoming>4</incoming>
</exclusiveGateway> </process> ... </bmn>
```

Figura 2. Exemplo de arquivo BPMN.

3. Metodologia

A metodologia adotada nessa pesquisa foi uma investigação aplicada. Em resposta à questão de pesquisa foi utilizado o método hipotético-dedutivo. Além disso, foi utilizado

um questionário semi-estruturado com escala *Likert*. A análise dos resultados não utilizou inferência estatística ou teste de hipótese.

Realizamos uma revisão da literatura em busca de algumas soluções adequadas para disseminar o uso de AIML chatbots por não programadores, avaliamos algumas ferramentas de construção de chatbots, porém, detectamos que o AIML carecia de recursos visuais que o mantivessem competitivos em relação a seus concorrentes comerciais. Apesar do melhor dos nossos esforços em procurar soluções visuais para criação de AIML chatbots não encontramos nenhuma que atendia aos nossos requisitos, entre eles: robustez, escalabilidade e possibilidade de reaproveitamento de ferramentas de edição já conhecidas pela comunidade. Portanto, a junção de dois grandes ecossistemas, com diversas ferramentas e bibliotecas disponíveis, foi a hipótese a ser confirmada.

Dividimos a validação do algoritmo conversor em duas etapas. Na primeira etapa (i) os próprios autores mapearam internamente alguns fluxos em BPMN2AIML definidos em outras pesquisas e disponibilizaram um desses chatbots para conversar com usuários, na segunda etapa (ii) os autores treinaram participantes na construção dos fluxos e coletaram dados com questionários para avaliar a percepção desses usuários. Para isso um treinamento no processo de desenhar os fluxos precisou ser ministrado, uma documentação de todo o processo de desenho também foi elaborada e compartilhada no treinamento. Esta documentação pode ser encontrada no “Apêndice A - DO DIAGRAMA AO DIÁLOGO” disponível em [Neo 2020].

3.1. Primeira etapa: Validação dos próprios autores

Para essa primeira etapa validamos a proposta criando dois chatbots, um que auxilia pessoas com depressão e um que ensina sobre ética. Fizemos uma revisão da literatura para buscar um chatbot com base relativamente complexa e com muitos fluxos descritos. Encontramos fluxos de diálogo de um chatbot sobre depressão. A partir desses fluxos já elaborados, mapeamos em diagramas BPMN2AIML, convertimos em AIML com o algoritmo de conversão proposto e carregamos o AIML gerado na base de conhecimento de um chatbot, por fim, conversamos com o chatbot construído. Porém, não disponibilizamos esse chatbot que auxiliava pessoas com depressão para usuários conversarem, pois não tivemos o apoio de profissionais da área e não nos sentimos seguros em prosseguir no tema.

Criamos outro chatbot no domínio da filosofia com conhecimento sobre ética. Um professor de filosofia realizou uma revisão da literatura sobre ética clássica, modelou os fluxos de diálogos necessários, em seguida os fluxos foram carregados no chatbot. Avaliamos com 81 estudantes como foi essa iteração com questionários. Detalhes adicionais relacionados a revisão da literatura está disponível em [Neo et al. 2020].

3.2. Segunda etapa: Validação com usuários

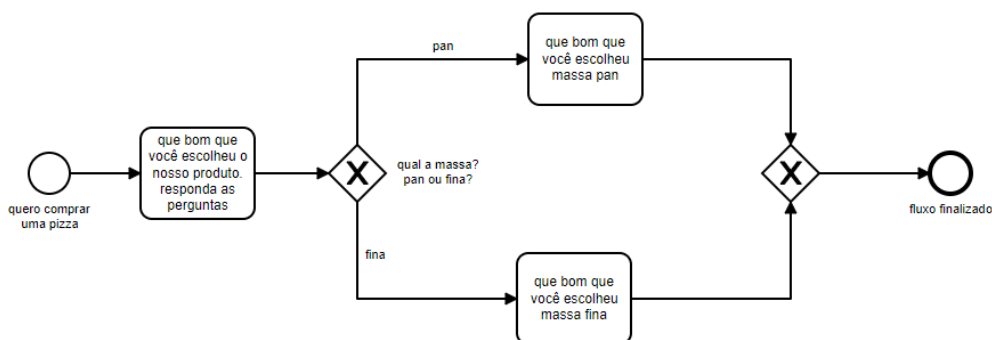
Para minimizar ameaças a reprodutibilidade da proposta e para minimizar os vieses dos pesquisadores realizamos também experimentos com participantes criando fluxos. Sugerimos que usuários criassem chatbots com temas sugeridos, que desenhassem seus próprios fluxos em ferramenta de modelagem BPM, que convertessem utilizando o conversor, e por fim, que carregassem os fluxos em um chatbot AIML. Avaliamos nesse ponto a usabilidade da solução ponta-a-ponta.

A nossa amostra total de participantes do experimento dessa etapa foi de 18 usuários. Foram 6 alunos do curso de ciência de computação que conheciam o AIML, eles participaram do experimento e responderam ao primeiro questionário. Em um segundo momento 12 pessoas da indústria e da academia, que não conheciam AIML, também participaram do mesmo experimento e responderam ao segundo questionário, que continham questões bem semelhantes. Ambos os questionários estão detalhados no “Apêndice D - QUESTIONÁRIOS” disponível em [Neo 2020].

4. Proposta: o BPMN2AIML

Para facilitar a construção de AIML chatbots e eliminar a necessidade do *botmaster* de conhecer XML na construção, propomos um conversor de BPMN para AIML, o BPMN2AIML.

Um exemplo de diagrama em BPMN, previamente desenhado em uma ferramenta visual de terceiros que suporta BPMN, mas pronto para a conversão pelo BPMN2AIML, pode ser visto na Figura 3. Esse seria um exemplo de entrada válida para o conversor. A partir deste diagrama, o conversor gera um arquivo AIML válido. Esse diagrama pode ser desenhado em qualquer ferramenta de desenho que suporte o padrão BPMN.



usuário: Quero comprar uma pizza
chatbot: Que bom que você escolheu o nosso produto. Responda às perguntas.
chatbot: Qual a massa? Pan ou Fina?
usuário: Pan.
chatbot: Que bom que você escolheu massa pan.
chatbot: Fluxo finalizado.

Figura 3. Acima diagrama simples na convenção BPMN2AIML. Abaixo uma conversa com um AIML chatbot carregado com o fluxo desenhado na sua base de conhecimento.

A arquitetura da solução buscou ser tão simples quanto possível. A partir do fluxo desenhado em qualquer ferramenta de diagramação BPMN é iniciado o procedimento. O conversor carrega esse fluxo em sua memória, em seguida faz um laço em todas as arestas (as setas do BPMN) e para cada uma cria as *tags* AIML ajustadas, finalizando com a saída de um único arquivo AIML.

O algoritmo BPMN2AIML (apresentado em Algoritmo 1) funciona da seguinte forma: para cada aresta do BPMN, desenhado conforme as convenções estabelecidas é

verificada um dos cinco condicionais (“se então”). Mais detalhes das convenções necessárias podem ser encontrados em “APÊNDICE A – DO DIGRAMA AO DIÁLOGO” [Neo 2020]. As condições envolvem verificar o elemento anterior e o elemento posterior da aresta, quando casar com uma das regras a condição é ativada; em seguida os valores das *tags* são atribuídos; por fim uma nova *tag category* contendo as *tags pattern*, *template* e *srai* que foram ativadas pela regra é criado, e o laço inicia novamente para a próxima aresta, até o fim das arestas do diagrama.

Os principais componentes BPMN suportados para mapear o fluxo de diálogo implementados foram: o evento de início; o *gateway* (classificados nesta pesquisa em 3 tipos: *gateway* simples, *gateway* com variável na anotação, *gateway* para *bypass*); a atividade; e o evento de fim. Um diagrama modelado em Dinâmica de Sistemas (SD) elaborado no Vesim que representa o fluxo desse processo de conversão está na Figura 4.

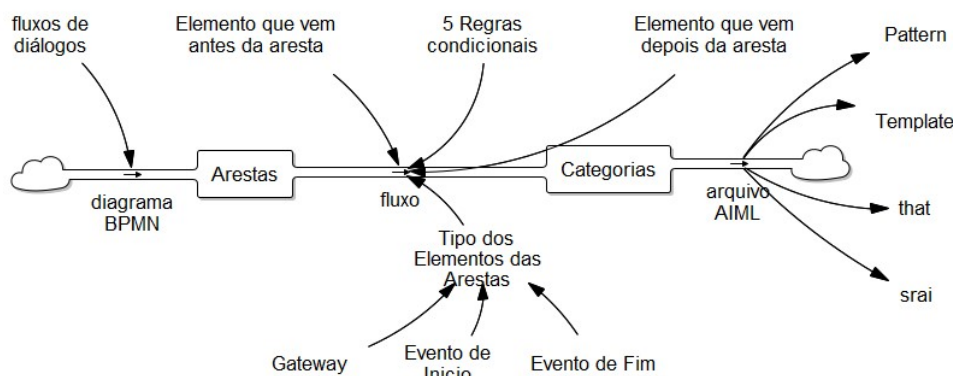


Figura 4. Representação em dinâmica de sistemas (SD) do algoritmo.

4.1. Prova de conceito: Implementação do algoritmo

Para validar os experimentos foi necessário construir uma prova de conceito com a implementação do algoritmo de conversão proposto e o chatbot. Fizemos um *fork* no *github* do *Program-y* [Sterling 2019], um interpretador *python* para o AIML 2.0. Implementamos o conversor nele e adaptamos a sua interface para carregar os AIML gerados pelo BPMN2AIML. Disponibilizamos todo o código-fonte para reprodução por outros pesquisadores do conversor e do chatbot no *github*² do projeto.

O canal de diálogo dos usuários com o chatbot selecionado foi o Facebook Messenger. Apesar do *Program-y* disponibilizar interface web para conversar com o chatbot, optamos por uma integração com a interface de acesso com o Facebook Messenger. Julgamos que uma interface já conhecida evitaria treinamentos adicionais dos participantes do experimento.

Para a integração do *backend* do chatbot com o Facebook Messenger hospedamos a nossa API no *Heroku*. O *Heroku* é um PaaS (plataforma como serviço) que permite aos desenvolvedores criar, executar e operar aplicativos inteiramente na nuvem [Raj 2019]. Para ser publicado como um chatbot totalmente automatizado no Facebook Messenger foi necessário atender à exigência de respostas em menos de 30 segundos, o que foi possível graças a velocidade do interpretador AIML adotado.

²https://github.com/giseldo/chatbot_BTA_BPMN_to_AIML/tree/master/bta

Algoritmo 1 Conversor de BPMN para AIML.

```
for aresta em arestas do  
  if aresta.ant é um evento de inicio then  
    pattern ← aresta.ant.nome  
    template ← aresta.pos.nome  
    srai ← aresta.pos.id  
  else if aresta.pos é um evento de fim then  
    pattern ← aresta.ant.id  
    template ← aresta.pos.nome  
  else if aresta.ant é uma atividade e aresta.pos é uma atividade then  
    pattern ← aresta.ant.id  
    template ← aresta.pos.nome  
    srai ← aresta.pos.id  
  else if aresta.ant é uma atividade e aresta.pos é um gateway then  
    pattern ← aresta.ant.id  
    template ← aresta.pos.nome  
  else if aresta.ant é um gateway then  
    that ← aresta.ant.nome  
    pattern ← aresta.nome  
    template ← aresta.pos.nome  
    srai ← aresta.pos.id  
  end if  
  nova categoria (that, template, pattern, srai)  
end for
```

5. Trabalhos relacionados

Uma comparação de algumas características de interesse com outras pesquisas é apresentado na Tabela 1. Grande parte da inspiração da proposta vem da pesquisa de [López et al. 2019]. Porém, existe uma diferença crucial entre as duas abordagens. Enquanto ele carrega todos os diagramas BPMN gerados em uma organização e gera um chatbot destinado a guiar o usuário mediante dúvidas sobre o processo de negócios modelado, a proposta do BPMN2AIML é representar a própria base de conhecimento do AIML chatbot modelado em BPMN.

Tabela 1. Comparação com outros trabalhos relacionados.

Pesquisa	P1	P2	P3	P4	P5
[Soares 2014]	NÃO	NÃO	SIM	NÃO	NÃO
[Rodríguez-Gil et al. 2019]	SIM	SIM	NÃO	NÃO	NÃO
[López et al. 2019]	SIM	SIM	SIM	SIM	NÃO
[Leopold et al. 2013]	SIM	SIM	SIM	SIM	NÃO
[Costa 2017]	SIM	SIM	SIM	SIM	NÃO
BPMN2AIML	SIM	SIM	SIM	SIM	SIM

Legenda:

P1 = Usado em chatbots?

P2 = Converte alguma modelagem visual em fluxo de diálogo?

P3 = Converte BPMN para algum propósito?

P4 = Usa AIML?

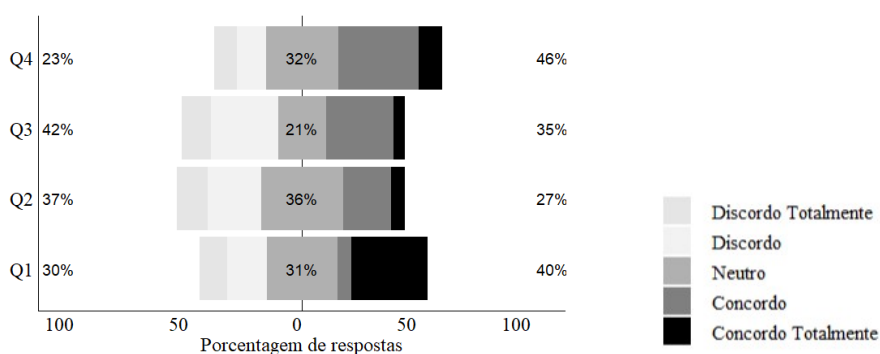
P5 = Converte o fluxo de diálogo de BPMN na base de conhecimento do AIML?

Uma solução comercial mais próxima do BPMN2AIML é a plataforma *Wit Conversation*. Ela compartilha uma semelhança no uso da notação visual BPMN para facilitar a construção de chatbots. Ela foi criada pela empresa *Wit-Software* [Costa 2017] e pretende simplificar a maneira como se constroem chatbots, nomeadamente nas suas integrações com *APIs* externas [Costa 2017]. Essa plataforma tem uma *engine* que representa os fluxos conversacionais que os chatbots podem possuir. Porém, não existe uma conversão direta do BPMN para o AIML, como acontece com o BPMN2AIML.

6. Resultados

Para avaliar a viabilidade do conversor escolhemos um chatbot para auxiliar pessoas com depressão baseado na teoria cognitivo comportamental delineado por [Junior 2017]. Essa pesquisa tem muitos fluxos replicáveis. Usamos estes diálogos apresentados no trabalho original em forma de texto e redesenhamos na nossa notação visual proposta utilizando o *Bizagi*. Esses fluxos visuais mapeados no *Bizagi* estão disponíveis no “Apêndice E - DIAGRAMAS BIZAGI” de [Neo 2020]. Dado a falta de um profissional da área para nos apoiar no tema optamos por não realizar testes conversacionais com usuários.

Realizamos mais um experimento com pessoas conversando com o chatbot, mas dessa vez conversando sobre o domínio da ética. Criamos este chatbot, com o apoio de um professor de filosofia. Textos fundamentais da ética foram selecionados pelo especialista; foi feito um mapeamento desse conhecimento ético e de possíveis fluxos conversacionais; os fluxos foram convertidos em AIML com a implementação do algoritmo conversor e carregado em um chatbot; por fim foi realizado um experimento em sala de aula onde 81 alunos conversaram com esse chatbot. Um questionário foi utilizado para avaliar a percepção dos estudantes. Os resultados estão concentrados na região central do gráfico (Figura 5). Portanto, não conclusivos em relação ao interesse, aprendizado, entendimento e possibilidade de recomendação do chatbot. Cabe ressaltar que a avaliação foi feita por pessoas que conversaram com o chatbot e não por quem utilizou o BPMN2AIML.



Legenda:

Q1 = Você considera que foi interessante conversar com o chatbot.

Q2 = Você aprendeu alguma coisa com essa conversa.

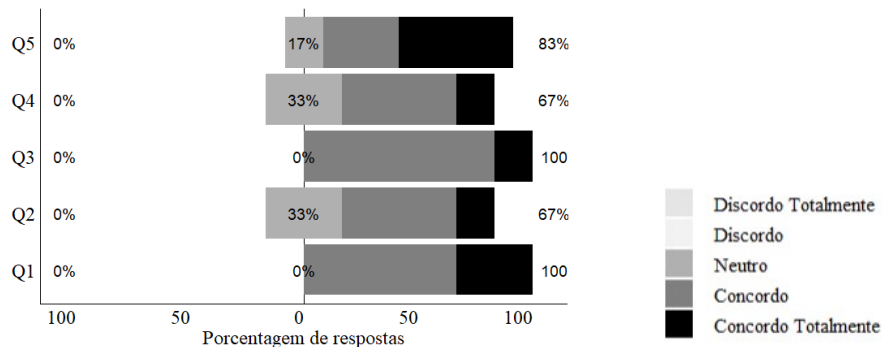
Q3 = O chatbot conseguiu entender as suas perguntas e respostas.

Q4 = Você conversaria novamente com o chatbot.

Figura 5. Respostas do questionário dos usuários que conversam com o chatbot.

Para verificar se recursos visuais do BPMN na construção dos fluxos de diálogo AIML facilitam a criação da base de conhecimento dos AIML chatbots, delineamos dois experimentos com participantes. Ao total 18 pessoas construíram chatbots utilizando o

BPMN2AIML. No primeiro experimento 6 pessoas com conhecimento em AIML (alunos do curso de ciência da computação) foram sugeridas a criar fluxos para chatbots com problemas de negócio sugeridos e em seguida carregar e conversar com o chatbot por eles mesmos criados. A descrição dos problemas pode ser vista em “APÊNDICE F – EXERCÍCIOS SUGERIDOS” de [Neo 2020]. Um resumo das respostas dos participantes desse primeiro experimento está na Figura 6. A maioria das respostas foi positiva, aparentemente confirmando a hipótese na amostra utilizada.



Legenda:

Q1 = Conheço AIML.

Q2 = Fiz o diagrama de fluxo de diálogo com BPMN2AIML em menor tempo do que com AIML diretamente.

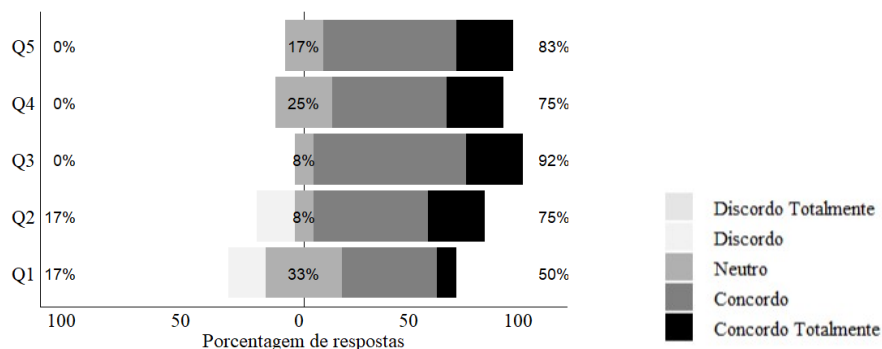
Q3 = Este software foi bem-sucedido no que ele se propõe a fazer em relação ao uso do AIML.

Q4 = Estou satisfeito com este software para utilizar em chatbots que interpretam AIML.

Q5 = Recomendaria para outra pessoa com conhecimentos em AIML esta ferramenta.

Figura 6. Respostas do questionário dos participantes que não conheciam AIML.

No segundo experimento 12 pessoas sem conhecimento em AIML passaram pelo mesmo processo, criaram fluxos e conversaram com o seu próprio chatbot. Um sumário das respostas dos participantes está na Figura 7.



Legenda:

Q1 = Não conheço AIML.

Q2 = Fiz o diagrama de fluxo de diálogo em pouco tempo.

Q3 = Este software foi bem-sucedido no que se propõe a fazer.

Q4 = Estou satisfeito com este software.

Q5 = Recomendaria para outra pessoa este software.

Figura 7. Respostas do questionário dos participantes que conheciam AIML.

A maioria das respostas, tanto do primeiro quanto do segundo experimento, aparentemente confirmaram que a percepção dos usuários em relação à construção dos chatbots foi bem-sucedida, que estavam satisfeitos e que recomendariam para outras pessoas.

7. Conclusão

Apresentamos um algoritmo que converte BPMN em AIML para facilitar a construção de fluxos de diálogo para ser utilizado em chatbots. Espera-se que o uso de chatbots seja cada vez mais ubíquo no nosso dia a dia. Lidar com *tags* XML tem suas vantagens, mas de forma geral fluxos em notação visual são mais interpretáveis por seres humanos, dado a capacidade humana em encontrar padrões visuais.

Encontramos indícios de aceitação da hipótese obtendo alta taxa de aprovação dos usuários que criaram seus próprios fluxos. Como a amostra foi selecionada por conveniência, mais estudos ainda são necessários para generalizar as conclusões. O resultado esperado é que não programadores possam criar AIML chatbots com uma curva de aprendizado menor, além disso, não seria necessário como pré-requisito conhecimento em XML. Uma contribuição desse estudo é um algoritmo para converter BPMN em AIML e a possibilidade de representação de toda a base de conhecimento de um AIML chatbot em BPMN. Algumas limitações foram identificadas: não foram contempladas todas as possibilidades que um diálogo complexo pode exigir; nem todos os símbolos BPMN foram suportados pelo conversor; e o suporte a caracteres especiais do texto não foi devidamente implementado.

Como trabalho futuro sugerimos avaliação de desempenho de velocidade do algoritmo; a integração de dinâmica de sistemas (SD) para a representação de ciclos de realimentação (*feedback loop*) do diálogo; a integração do *Decision Model and Notation* (DMN) na definição das regras mais trabalhosas do fluxo; a implementação de técnicas de PLN para classificar intenções; e estender a notação para incluir outros conceitos, tais como: resposta randômica. Outra possibilidade é gerar a base de conhecimento visual a partir de algoritmos com aprendizagem de máquina; converter regras de decisão a partir de árvores de decisão em fluxos visuais; por fim, mais estudos de caso em outros domínios e experimentos controlados são necessários para generalizar as conclusões.

Referências

- ABPMP, I. (2013). *ABPMP BPM CBOK V3.0 Guide to the Business Process Management Common Body of Knowledge*.
- AISB (2019). Mitsuku wins 2019 Loebner Prize and Best Overall Chatbot at AISB X.
- Barros, F. d. A. and Tedesco, P. A. (2016). Agentes inteligentes conversacionais: Conceitos básicos e desenvolvimento. *35º JAI-Jornada de Atualização em Informática*, pages 169–218.
- Borah, B., Pathak, D., and Sarmah, P. (2019). Survey of Textbased Chatbot in Perspective of Recent Technologies. In *CICBA 2018: Computational Intelligence, Communications, and Business Analytics*, volume 1031, pages 84–96. Springer Singapore.
- Costa, J. C. G. (2017). *Assistentes Virtuais para Comunicação Empresarial*. PhD thesis.
- Dumas, M., La Rosa, M., Mendling, J., and Reijers, H. A. (2013). *Fundamentals of Business Process Management*.
- Junior, O. A. d. A. (2017). *Beck: Um Chatbot Baseado na Terapia Cognitivo-Comportamental para Apoiar Adolescentes com Depressão*. Dissertação de mestrado, UFPE, Recife - PE - Brasil.

- Larisane Kuyven, N., André Antunes, C., João de Barros Vanzin, V., Luis Tavares da Silva, J., Loureiro Krassmann, A., and Margarida Rockenbach Tarouco, L. (2018). Chatbots na educação: uma Revisão Sistemática da Literatura. *Renote*, 16(1):123–132.
- Leopold, H., Mendling, J., and Polyvyanyy, A. (2013). Supporting Process Model Validation through Natural Language Generation. *Handbook of Natural Language Processing, Second Edition*, X(X):121–144.
- López, A., Sánchez-Ferreres, J., Carmona, J., and Padró, L. (2019). From Process Models to Chatbots. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11483 LNCS:383–398.
- Neo, G. d. S. (2020). *Construção de chatbots AIML com a ajuda de uma ferramenta de modelagem visual baseada na linguagem BPMN*. Dissertação de mestrado, UFAL, Maceió - AL - Brasil.
- Neo, G. S., Pereira, O. M., and Neo, A. V. B. S. (2020). Engajando alunos na aula de filosofia com inteligência artificial. In *VII Semana Internacional de Pedagogia*, number 1, pages 1–12, Maceió - AL - Brasil.
- Oladele, M. I., Sunar, M. S. B., Cheok, A. D., Arovski, S., and Osipyan, H. (2017). Visual Dialog Manager: a Tool for Authoring Conversational Agents. *International Conference on Sciences, Engineering and Social Sciences*, (September).
- OpenAI (2023). GPT-4 Technical Report. 4:1–100.
- Raj, S. (2019). *Building Chatbots with Python*.
- Rodríguez-Gil, L., García-Zubia, J., Orduña, P., Villar-Martinez, A., and López-De-Ipiña, D. (2019). New Approach for Conversational Agent Definition by Non-Programmers: A Visual Domain-Specific Language. *IEEE Access*, 7:5262–5276.
- Shum, H.-y., He, X.-d., and Li, D. (2018). From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.
- Silveira, C. (2018). Easy AIML - Gerador de códigos AIML. *Revista CNEC Educação*, pages 86–99.
- Soares, J. J. M. (2014). *Uma abordagem para a transformação da notação bpmn para a notação de redes de petri*. PhD thesis.
- Sterling, K. (2019). Python 3.x based aiml 2.0 chatbot interpreter, framework. <https://github.com/keiffster/program-y>. Accessed: 2019-11-17.
- Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, pages 551–621.
- Wallace, R. S. (2009). The anatomy of A.L.I.C.E. *Parsing the Turing Test*. Springer, Dordrecht, pages 181–210.
- Weizenbaum, J. (1966). ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Communications of the ACM*, 9(1).