

Raciocínio Causal em Agentes BDI: um Modelo Abstrato

João Faccin¹, Ingrid Nunes^{1,2}

¹Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil

²TU Dortmund
Dortmund, Germany

{jgfacin, ingridnunes}@inf.ufrgs.br

Abstract. Reasoning about cause-effect relationships is a recurrent step in the human decision-making process. It is quite common for us to face problems that are consequences of other problems, and that can only be considered solved when their causes are also addressed. In Computer Science, several instances of this scenario can be identified. Although existing proposals are able to solve consequences as well as to address causes appropriately, the reasoning about causal relationships is manually implemented and application-specific. In this paper, we discuss the use of software agents as a feasible alternative to address such scenario and present a proposal toward a domain-independent solution using the Belief-Desire-Intention (BDI) architecture.

Resumo. Raciocinar sobre relações de causa e efeito é um passo recorrente no processo humano de tomada de decisão. Não é raro que enfrentemos problemas que são consequências de outros problemas, e que só podem ser considerados solucionados quando suas causas também o forem. Em Ciência da Computação, diversas instâncias desse cenário podem ser identificadas. Apesar de propostas existentes serem capazes de solucionar consequências, bem como endereçar causas adequadamente, o raciocínio sobre relações causais é implementado manualmente e de forma específica para cada aplicação. Neste artigo, discutimos o uso de agentes de software como uma alternativa possível capaz de lidar com tal cenário, e apresentamos uma proposta para uma solução independente de domínio utilizando a arquitetura BDI (Belief-Desire-Intention).

1. Introdução

Damos o nome de *causalidade* ao relacionamento de *causa* e *efeito* entre eventos ou fatores [Pearl 2009]. Uma causa, nesse contexto, é um evento que antecipa outro, de modo que o último não ocorre caso o primeiro não aconteça. Um efeito, por outro lado, é aquele evento cuja causa reside em outro evento. Tais relações causais possuem um papel fundamental no modo como os seres humanos estruturam o mundo, pois é através delas que buscamos explicar nossas observações e é sobre elas que fundamentamos nosso conhecimento científico. Quando avistamos um gramado molhado, por exemplo, podemos presumir que seu estado seja resultado, i.e. um efeito, de uma chuva ou irrigação recente.

Entretanto, mais do que apenas um recurso explicativo, as relações de causalidade são elemento crucial no processo de raciocínio prático e tomada de decisão, especialmente quando lidamos com problemas cuja solução definitiva envolve a resolução de suas

causas. Nesse tipo de situação, o problema inicial, aqui visto como um efeito, deve ser mitigado, quando os efeitos são indesejados e precisam ser endereçados antes mesmo que uma solução definitiva seja encontrada. Posteriormente, os fatores que compõem sua causa são identificados e tratados. Ao final deste processo, tendo todos os fatores que compõem a causa como resolvidos, o problema inicial terá sido definitivamente solucionado. Exemplos relacionados a este cenário não são raros em nosso cotidiano. Considere a seguinte situação: ao chegar em casa, você se depara com uma poça d'água no chão da sala de estar e, imediatamente, percebe que a mesma surgiu devido a uma goteira no teto do cômodo. A fim de remediar a situação, você insere um balde sob a goteira, impedindo o aumento da poça d'água. Com o problema inicial sob controle, você se volta à identificação e solução das causas do mesmo — uma goteira pode ser resultado de um vazamento no encanamento, uma telha quebrada, entre outros fatores. Ao solucionar as causas da goteira, seus efeitos cessarão, o balde poderá ser removido e o problema se dará por resolvido. Apesar de simples, esse exemplo demonstra como o raciocínio causal influencia o modo como agimos. Se as causas da goteira fossem ignoradas, recursos e esforço excessivos seriam necessários a fim de evitar o aumento da poça d'água. Por outro lado, se tais causas fossem consideradas antes de seus efeitos serem mitigados, as consequências seriam agravadas. No nosso exemplo, a poça d'água aumentaria consideravelmente enquanto as causas da goteira fossem identificadas e tratadas.

Em Ciências da Computação, instâncias deste cenário podem ser encontradas nos mais diversos domínios, de sistemas adaptativos a redes de computadores e *smart grids*, entre outros. Propostas existentes são capazes de solucionar consequências e endereçar causas adequadamente [Goldstein et al. 2007, Schaeffer-Filho et al. 2012]. Entretanto, esse raciocínio sobre causas e efeitos é implementado de forma manual e específica para cada cenário, limitando a possibilidade de reuso. Neste trabalho, discutimos como o uso de agentes de software pode ser visto como uma alternativa apropriada para a implementação dessa estratégia de resolução de problemas, assim como os esforços necessários para o desenvolvimento de uma solução independente de domínio. Na Seção 2 exploramos os diferentes aspectos do problema abordado. Uma proposta para o desenvolvimento de uma solução baseada em agentes é discutida na Seção 3. Finalmente, trabalhos relacionados são descritos na Seção 4, e conclusões apresentadas na Seção 5.

2. Problema

Raciocinar sobre problemas que envolvem causalidade é, a primeira vista, trivial para os seres humanos. Porém, essa visão muda quando quem realiza tal raciocínio passa a ser um sistema computacional. Considerar relações causais na resolução de problemas a partir de soluções de software envolve dois questionamentos principais. O primeiro deles diz respeito a *como representar* tais relações. Apesar do conjunto de relacionamentos de causa e efeito que caracteriza um domínio variar de acordo com a pessoa que o modela, é necessário que se estabeleçam critérios que permitam que tal modelagem seja realizada de maneira padronizada. De fato, diversos trabalhos propõem modelos causais que fornecem meios de descrever essas relações, desde maneiras gráficas [Pearl 1995] até aquelas que envolvem modelos matemáticos e estatísticos [Halpern 2015]. Dadas suas características individuais, alguns desses modelos são mais informativos, e conseqüentemente mais complexos que outros.

O segundo questionamento se refere a *como raciocinar* efetivamente sobre um

modelo causal fornecido. A busca por uma resposta a essa questão traz à tona diversos desafios. Consideremos novamente o cenário ilustrativo apresentado na seção anterior, onde é necessário lidar com uma poça d'água na sala de casa. Para endereçar o tipo de cenário proposto, é necessário que seja possível determinar se o problema considerado é de fato um efeito de outro(s) problema(s), ou seja, se a poça d'água tem origem em algum outro evento. Em caso afirmativo, é necessário considerar que sua causa pode ou não ser conhecida. No nosso exemplo, a poça d'água acontece devido a uma goteira, que, por sua vez, pode ter origem em um vazamento no encanamento ou em uma telha quebrada em um dia chuvoso. Apesar de podermos definir a goteira como um efeito, não é possível estabelecer imediatamente quais ou quantos dos fatores citados são de fato a causa do problema. Dessa forma, um sistema apto a lidar com esse cenário deve ser capaz não apenas de determinar possíveis fatores de uma causa, mas também identificar aqueles que efetivamente compõem a causa em um determinado momento.

Outros desafios surgem a partir do instante em que ações são tomadas para que causas e efeitos sejam solucionados. No nosso exemplo, um balde é colocado sob a goteira a fim de mitigar o problema inicial, e a partir daí sua causa é identificada e tratada. Um sistema que lide com tal cenário deve ser capaz de determinar quando os fatores causadores do problema inicial foram solucionados e, assim, estabelecer que tal problema pode ser dado como definitivamente resolvido. Adicionalmente, deve-se considerar a necessidade de *reverter ações* tomadas no processo de remediação do problema inicial. Em nosso cenário ilustrativo, por exemplo, não é necessário que se mantenha o balde na sala após o fim da goteira.

Por fim, em propostas existentes capazes de solucionar problemas endereçando suas causas [Goldstein et al. 2007, Schaeffer-Filho et al. 2012], todo o raciocínio causal é implementado manualmente. Assim, cabe aos desenvolvedores processar as relações de causa e efeito existentes entre os diversos fatores que compõem o domínio e determinar de maneira programática as sequências de ações a serem executadas quando houver problemas que requeiram uma solução. Dessa forma, tais soluções acabam tornando-se específicas para cada domínio. Além disso, muito esforço é exigido se considerarmos a necessidade de adicionar ou remover relações causais, tornando esta uma tarefa demorada e propensa a erros. Essa série de questionamentos e desafios, além da existência de diversas instâncias do cenário proposto nos mais diversos domínios, considerando principalmente a crescente autonomia que software está recebendo, nos mostra que o processo de resolução de problemas baseado no raciocínio causal é um ponto que possui um grande potencial de pesquisa. A seguir, discutimos como a tecnologia de agentes de software, em especial de agentes baseados na arquitetura BDI, pode ser aproveitada nesse cenário.

3. Agentes de Software como Alternativa

O uso de agentes de software surge como uma alternativa flexível para o desenvolvimento de aplicações capazes de lidar com o cenário considerado, especialmente em domínios de grande complexidade. Na prática, abordagens existentes já tiram proveito dessa flexibilidade, utilizando agentes como meio de resolver consequências e solucionar suas causas [Nunes et al. 2017]. Entretanto, mesmo nestes trabalhos, o raciocínio sobre causas e efeitos ainda é implementado de forma manual. Nossa proposta explora a arquitetura BDI como base para a criação de um modelo de agente apto a lidar com os problemas discutidos na Seção 2, ao mesmo tempo em que abstrai especificidades do domínio no

processo de raciocínio causal e resolução de problemas.

A arquitetura BDI (do inglês *belief-desire-intention*) [Rao and Georgeff 1995] estrutura agentes em termos de atitudes mentais de crenças, desejos e intenções. Crenças representam a informação que o agente possui a respeito do mundo, enquanto desejos (também chamados de objetivos) são os estados de mundo que este agente deseja alcançar. Intenções, por sua vez, dizem respeito ao comprometimento do agente em atingir determinados objetivos. Agentes BDI têm seus objetivos explicitamente definidos, e planos para atingí-los são previamente especificados. Tais agentes executam um ciclo de raciocínio que compreende diversas funções abstratas que manipulam suas atitudes mentais. Uma delas é a *função de geração de opções*, responsável por manter o conjunto atualizado de desejos do agente, inserindo ou removendo objetivos conforme necessário. Em nossa proposta, tal função de geração de opções é customizada de modo a refletir a estratégia de resolução de problemas baseada no raciocínio causal.

Essa estratégia, descrita através de exemplos nas Seções 1 e 2, pode ser visualizada como uma série de passos a serem executados da seguinte forma.

1. Dado um problema, executam-se ações para resolvê-lo enquanto verifica-se a existência de possíveis causas para o mesmo;
2. Se possíveis causas forem encontradas, determina-se quais delas são de fato causadoras do problema naquele momento;
3. Identificadas as causas atuais, instancia-se cada uma delas como um problema a ser resolvido;
4. Quando o problema e suas respectivas causas forem solucionados, o problema inicial é considerado definitivamente resolvido.

Ao realizarmos uma correspondência entre problemas e os objetivos especificados na arquitetura BDI, percebemos que a execução desses passos resume-se à busca por relações causais entre objetivos e à adição e remoção destes objetivos do conjunto atualizado de desejos do agente. Podemos notar também que tal sequência de passos não depende de nenhuma informação específica de domínio além do conjunto de relações de causa e efeito do mesmo. A fim de resolvermos essa questão e termos um modelo que seja de fato independente de domínio, propomos o desacoplamento do modelo causal da função de geração de opções. Desse modo, a obtenção de informações referentes aos relacionamentos entre problemas passa a ser responsabilidade do próprio modelo causal, sendo delegada à função de geração de opções apenas a tarefa de consultar tais informações. Adicionalmente, representar o conjunto de relações de causa e efeito de modo independente tende a reduzir o esforço exigido por parte dos desenvolvedores em tarefas como a inserção ou remoção de relações do conjunto. Entretanto, como mencionado na Seção 2, existem vários trabalhos sugerindo diferentes modelos causais com graus de informação e complexidade distintos, os quais detalharemos na próxima seção. Uma investigação a respeito da conveniência do uso destes modelos em nossa proposta ou da necessidade de desenvolvimento de um modelo causal específico é, portanto, um dos passos essenciais a serem conduzidos no contexto da abordagem a ser elaborada. Finalmente, a Figura 1 ilustra nossa proposta, representando o relacionamento entre os elementos e funções da arquitetura BDI customizada e o modelo causal independente.

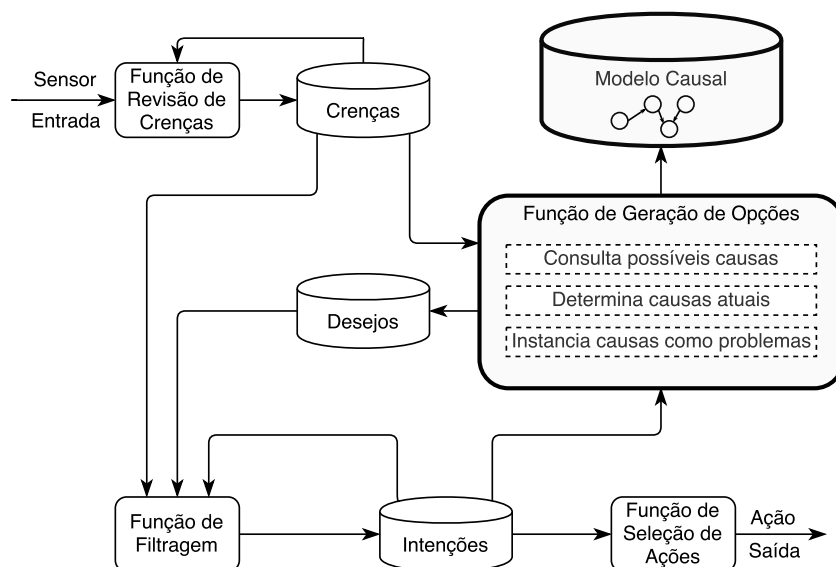


Figura 1. Relacionamento entre Elementos da Arquitetura BDI Customizada.

4. Trabalhos Relacionados

Até o momento da escrita deste artigo, os autores desconhecem outros trabalhos que abordem o cenário de resolução de problemas considerando relações causais de modo similar ao proposto. Entretanto, diversas propostas existentes podem ser consideradas complementares à nossa. Nas implementações típicas de agentes BDI, um objetivo pode ter apenas dois estados: totalmente atingido ou não satisfeito. Porém, essa representação binária é uma simplificação que, na maioria dos casos, não se sustenta em cenários reais onde um objetivo tende a ser atingido no decorrer do tempo. No cenário que motiva este trabalho, por exemplo, um problema estará parcialmente resolvido quando for remediado mas suas causas ainda não estiverem solucionadas.

Van Riemsdijk e York-Smith (2012) abordam este problema apresentando um arcabouço abstrato que pode ser usado como base para a representação da satisfação parcial de objetivos em agentes. Esse arcabouço permite a representação de noções quantitativas de parcialidade por meio do uso de métricas de progresso. Dessa forma, um agente consegue avaliar o quão próximo ele está de atingir seu objetivo. Entretanto, os autores não fornecem detalhes sobre mecanismos para computar tal representação. Essa questão é resolvida por Thangarajah et al. (2014) ao apresentarem uma abordagem que pode ser utilizada computacionalmente para quantificar o nível de completude de um objetivo baseando-se em anotações relacionadas ao conjunto de planos e objetivos do agente.

No que diz respeito à representação de relações de causa e efeito, muitos trabalhos propõem modelos causais de propósito geral. O uso de modelos gráficos sugerido por Pearl (1995) baseia-se na teoria dos grafos. Nesta representação, nodos simbolizam fatores ou eventos, e arestas direcionadas indicam relacionamentos causais. Este tipo de representação, mapeado para estruturas de dados específicas, é uma alternativa para uso em nossa proposta. O modelo de equações estruturais [Pearl 2009, Halpern 2015], por outro lado, não possui uma representação gráfica específica. Este modelo caracteriza o mundo por meio de variáveis e seus valores, e a influência entre variáveis — e a sua consequente relação causal — é representada por meio de equações. Apesar de um maior

poder informativo, este modelo carrega consigo uma maior complexidade.

5. Conclusão

Muitas situações do mundo real requerem que um efeito seja remediado enquanto suas causas são endereçadas, sendo esse raciocínio sobre causalidade um elemento frequente no processo de tomada de decisão e resolução de problemas. Existem diversas propostas abordando esse cenário e fornecendo soluções específicas para determinados domínios. Neste trabalho, discutimos os esforços iniciais para o desenvolvimento de uma abordagem independente de domínio baseada na arquitetura BDI. Em nossa proposta, a representação de relacionamentos causais é desacoplada do processo de raciocínio, tornando possível a reutilização desse mesmo processo em diferentes domínios. Por apresentarmos uma visão inicial da solução, muitos dos questionamentos levantados no decorrer do texto permanecem em aberto, como a necessidade de manter o estado das causas de um problema e a possibilidade de desfazer ações remediadoras após um objetivo ser atingido, por exemplo. A busca por tais respostas é objeto de trabalhos futuros.

Referências

- Goldstein, M., Shehory, O., and Weinsberg, Y. (2007). Can self-healing software cope with loitering? In *Fourth International Workshop on Software Quality Assurance: In Conjunction with the 6th ESEC/FSE Joint Meeting*, SOQUA '07, pages 1–8, New York, NY, USA. ACM.
- Halpern, J. (2015). A modification of the Halpern-Pearl definition of causality. *IJCAI 2015*, pages 3022–3033.
- Nunes, I., Schardong, F., and Schaeffer-Filho, A. (2017). BDI2DoS: an application using collaborating BDI agents to combat DDoS attacks. *Journal of Network and Computer Applications*.
- Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, 82(4):669–688.
- Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition.
- Rao, A. S. and Georgeff, M. P. (1995). BDI agents: From theory to practice. In *First International Conference on Multi-Agent Systems (ICMAS)*, pages 312–319.
- Schaeffer-Filho, A. E., Smith, P., Mauthe, A., Hutchison, D., Yu, Y., and Fry, M. (2012). A framework for the design and evaluation of network resilience management. In *NOMS*, pages 401–408. IEEE.
- Thangarajah, J., Harland, J., Morley, D. N., and Yorke-Smith, N. (2014). Quantifying the completeness of goals in BDI agent systems. In *ECAI - 21st European Conference on Artificial Intelligence*, pages 879–884.
- van Riemsdijk, M. B. and Yorke-Smith, N. (2012). Towards reasoning with partial goal satisfaction in intelligent agents. In *Proceedings of the 8th International Conference on Programming Multi-Agent Systems*, ProMAS'10, pages 41–59, Berlin, Heidelberg. Springer-Verlag.