

3

Análise de Criticalidade de Agentes: Fundamentos e Trabalhos Relacionados

Para estimar a criticalidade de um agente, é necessário entender que tipo de informações seria pertinente, e como obtê-las. Dentre as formas de obtê-las, temos, por exemplo, as informações especificadas pelo projetista do sistema, informações inferidas por um observador externo aos sistemas funcionais do agente, a quantidade de mensagens trocadas entre os agentes, uma observação interna do comportamento do agente, os planos do agente, e, finalmente, as leis que regulam as interações entre agentes.

Nas seguintes subseções, são apresentadas as diversas estratégias existentes na literatura para estimar a criticalidade de um agente. Algumas estratégias são genéricas e utilizam informações básicas, tais como mensagens e referências. Outras estratégias consideram informações de abstrações de mais alto nível, tais como, performativas e papéis que os agentes desempenham em uma determinada organização e planos.

3.1. Dependências Estáticas

O conceito de dependência está relacionado com o envolvimento entre agentes, i.e., quais agentes são dependentes de um outro agente. Intuitivamente, quanto maior o número de agentes dependentes de um determinado agente, mais crítico o último é para a organização.

Neste contexto, grafos [1] foram propostos com o objetivo de descrever uma forma de especificar as interdependências entre agentes. E o trabalho [3] utilizou esta abordagem para estimar e inferir tais dependências de forma automática. Uma primeira estimativa das dependências poderia ser feita estatisticamente usando instruções de envio de mensagem no código fonte dos agentes.

Começando pelo código fonte de todo o sistema multiagente, instruções de envio de mensagens são extraídas automaticamente. Nesta abordagem, assume-se

que o comportamento do agente é estruturado através de um conjunto de regras de transição <condição, ação>. Assim, o mecanismo de extração de código pode usar informações de causalidade sobre a condição (recebimento de mensagem) e a ação (envio de mensagem). Foram utilizados três algoritmos para gerar os grafos e inferir sobre as informações extraídas dos gráficos à fim de estimar a criticalidade dos agentes.

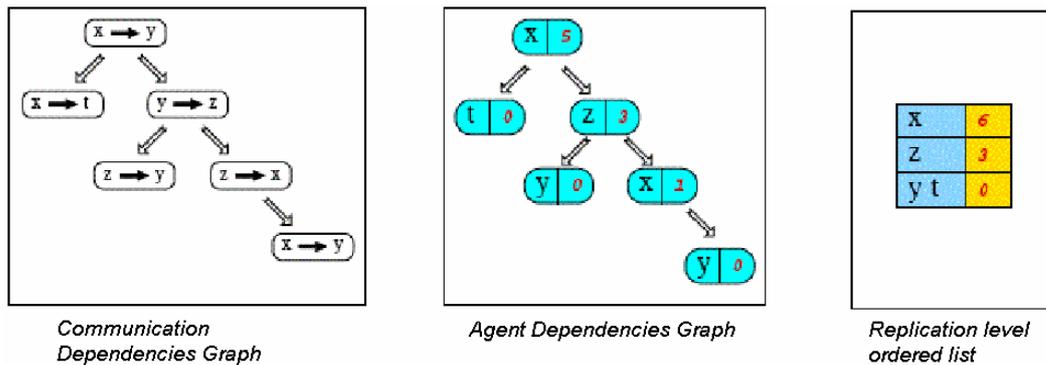


Figura 7 - Extração Estática de Dependências [25]

O primeiro algoritmo proposto computa automaticamente o grafo de dependência de comunicação. De forma ilustrativa, um grafo de dependência é exibido no lado esquerdo da Figura 7, onde cada nó é uma expressão de comunicação ($x \rightarrow y$ significa que o agente x envia uma mensagem para o agente y). Um segundo algoritmo proposto transforma o grafo de dependência de comunicação em um grafo de dependência de agentes (elemento do meio da Figura 7), e cada nó corresponde a um agente e um rótulo indicando quantos agentes são seus dependentes (i.e, podem receber mensagens deste agente). E finalmente, o último algoritmo proposto entra as partes conexas do grafo, árvores de cobertura mínima, e uma lista ordenada decrescentemente com a criticalidade dos agentes (informação exibida no lado direito da Figura 7).

A principal limitação desta estratégia é sua característica estática no sentido que de que o código fonte dos agentes é considerado e não a comunicação emergente das interações. Logo só é possível capturar comunicações em potencial e não se garante a captura das comunicações que de fato ocorreram.

3.2. Dependências Dinâmicas

Nesta estratégia [2], o problema do monitoramento de estruturas dinâmicas e emergentes é resolvido através da representação explícita das dependências entre os agentes como um grafo de pesos, provendo um mecanismo automático de atualizar seus respectivos pesos de acordo com as comunicações entre os respectivos agentes.

Desta forma, eles definem um grafo de interdependência como um grafo orientado a rótulos (veja Figura 8), onde cada nó representa um agente do domínio e cada arco rotulado representa a dependência entre os agentes associados. O rótulo de um arco orientado é um número real que reflete a importância da dependência (orientada) entre os agentes associados. O grafo de interdependência é dinâmico e pode ser modificado quando um novo agente do domínio é adicionado no sistema, quando um agente é retirado e quando o padrão de iteração dos agentes evolui.

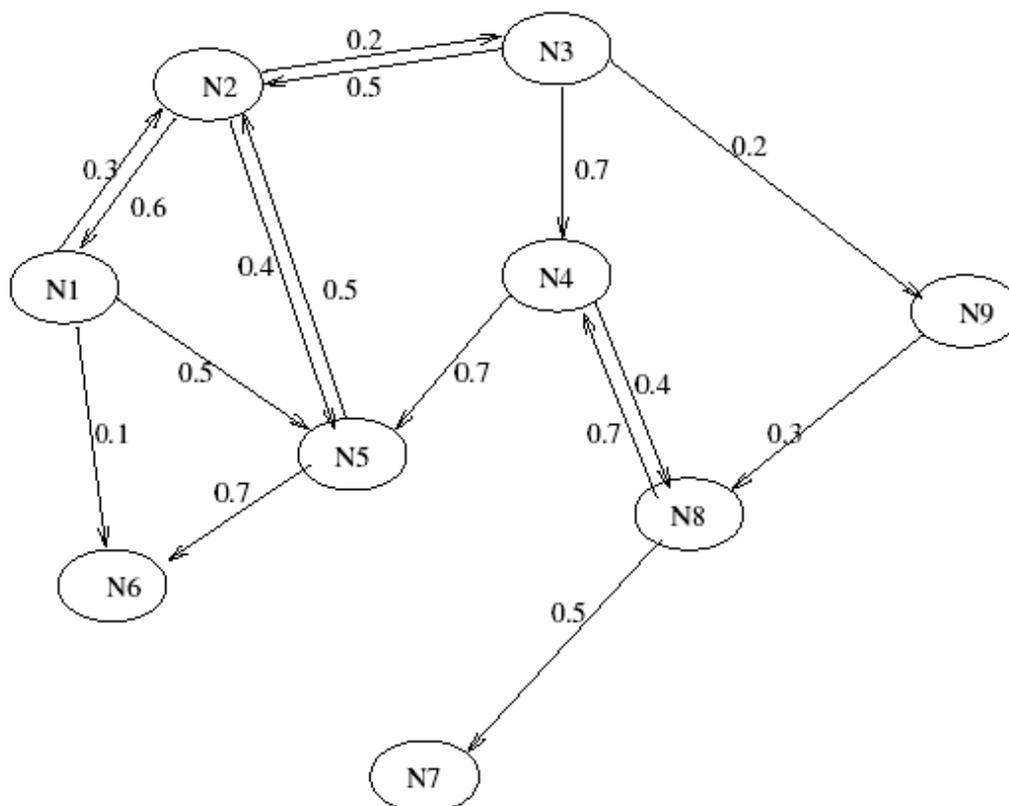


Figura 8 - Exemplo de Grafo de Interdependência

Em tempo de projeto, o grafo de interdependência é inicializado pelo projetista, e em tempo de execução, ele é adaptado dinamicamente. Dentre os

parâmetros utilizados para atualizar as interdependências entre os agentes, existe o parâmetro de carga de comunicação, que considera o número de mensagens.

Esta abordagem propôs dois algoritmos de adaptação: um considera somente o número de mensagens, independente do seu conteúdo. E o outro é uma extensão do primeiro e considera as performativas das mensagens como informações de entrada adicionais. Desta forma, por exemplo, a performativa *request* terá um peso maior que a performativa *cancel*.

Apesar de esta abordagem resolver o problema de dependências dinâmicas, ela não leva em consideração características do domínio da aplicação, a não ser as performativas das mensagens. Porém, alguns sistemas necessitam que certas informações específicas do domínio, e que conseqüentemente tornam o monitoramento da criticalidade algo de fundamental importância, sejam incorporadas nessa análise. Tornando o mecanismo apresentado nesta seção incompleto e insuficiente.

3.3. Papéis

Existe ainda uma terceira estratégia [4] baseada no conceito de papel. Um papel representa um padrão de serviços, atividades e relacionamentos que um agente pode desempenhar dentro de uma organização. Em algumas organizações de comércio eletrônico, por exemplo, os papéis seriam provedor de serviço, cliente, corretor, etc. Um papel pode ser desempenhado por um ou mais agentes e, ao mesmo tempo, que um agente pode desempenhar mais de um papel em diferentes organizações. Em uma aplicação de educação à distância, por exemplo, existiria um agente pessoal que ora desempenharia o papel de estudante, ora o papel do professor, ora o papel de coordenador.

Papéis podem não somente ser definidos em um contexto organizacional como na definição de protocolos de interação das organizações. Como um exemplo deste último temos o protocolo *Contract Net Protocol* [5], que considera dois papéis a serem desempenhados pelos agentes: o gerente, que é responsável por enviar mensagens de chamadas de propostas para n agentes, e o licitante que lança propostas.

Segundo esta abordagem, a noção de papel captura informações de importâncias relativas aos papéis e suas interdependências, e é utilizada para estimar a criticalidade dos agentes que desempenham tais papéis.

Para utilizar esta abordagem, o projetista precisa especificar um valor de importância para cada papel das organizações do sistema. E este valor deve estar no intervalo $[0,1]$. Não é necessário que o sistema implemente um mecanismo de entrada e saída em cada organização para o agente. Existe um mecanismo de monitoramento de papéis que usa uma linguagem declarativa para especificar os protocolos e um algoritmo de reconhecimento. Desta forma, a criticalidade de um agente é calculada como uma agregação de pesos dos conjuntos de papéis que estão sendo desempenhados no momento.

Esta abordagem é muito interessante e inicia um estudo de análise de criticalidade a partir do domínio da aplicação. Quando combinada com a estratégia anterior, pode gerar um monitoramento bastante interessante. Entretanto, como mencionado anteriormente, continua sendo uma estratégia incompleta e insuficiente visto que o conceito de papéis não cobre todos os riscos e que influenciariam a criticalidade dos agentes.

3.4. Planos

Todas as estratégias apresentadas até o momento são independentes da implementação dos agentes. Entretanto, existe ainda uma abordagem proposta em [6] baseada nos planos dos agentes, i.e., as ações para as quais um agente foi planejado a fazer em um futuro próximo.

Neste modelo, considera-se que cada agente do sistema conhece que seqüência de ações (seu plano) deve ser executada para atingir seus objetivos. E assume-se que dado um instante do tempo, o agente está executando no máximo uma ação. Desta forma, o plano de um agente é representado em um grafo AND/OR acíclico direcionado onde cada nó representa uma ação e os nós são conectados por um arco AND ou OR.

Na

Figura 9, por exemplo, após desempenhar a ação A, o agente Agent1 precisa executar as ações B e C a fim de seguir o seu plano. Contudo, após executar C,

somente a ação D ou E precisam ser executadas para que o agente cumpra com o seu plano. Esta proposta define ainda ações externas e ações terminais. Uma ação externa é aquela que um outro agente precisa executar para que o plano de um determinado agente seja cumprido. E uma ação terminal é aquela na qual nenhuma outra ação será executada para que o agente cumpra seu plano.

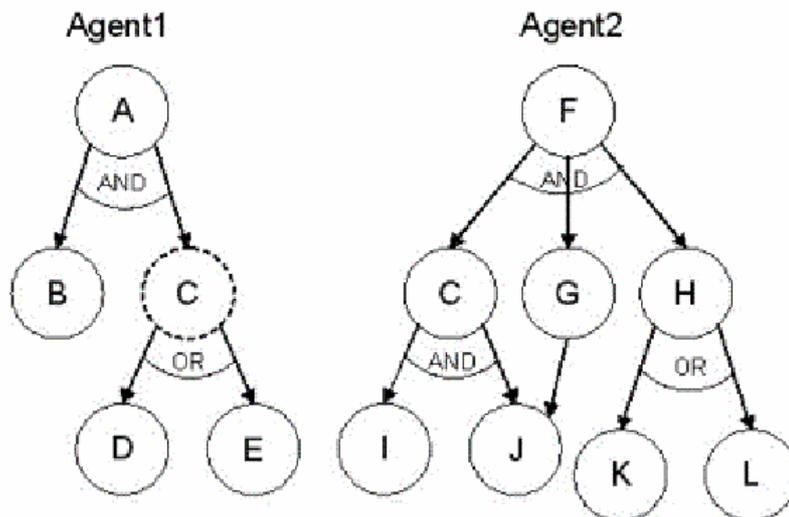


Figura 9 - Exemplo de Planos [25]

Para calcular a criticalidade de uma ação, esta abordagem diferencia sua criticalidade absoluta da relativa. A criticalidade absoluta (CA) de uma ação é definida sem levar em consideração os planos correntes dos agentes. Ela é dada *a priori* pelo projetista do sistema e pode ser determinada em função do número de agentes capazes de desempenhar a ação, em função da duração da ação, em função dos recursos necessário para a execução da ação, e em função das informações dependentes da aplicação.

A criticalidade relativa (CR) de uma ação pertencente ao plano de um agente é proporcional à criticalidade de um agente quando este está executando a ação ou esperando por um agente que a execute. Como consequência, a CR de uma ação varia de acordo com o plano a qual pertence. A CR é calculada da seguinte forma:

- ✓ Para uma ação externa, CR é igual a criticalidade relativa local (CRL). A CRL é obtida usando uma função de agregação do operador AND se a ação estiver conectada com seus filhos por meio de arcos AND, ou usando uma função de agregação do operador OR, se a ação estiver conectada com seus filhos por meio de arcos OR. Se a ação possui

somente um filho, sua CRL é igual à criticalidade relativa de seu filho.

Se a ação é terminal, sua CRL é igual a 0.

- ✓ Para uma ação não externa, sua CR é igual a sua criticalidade absoluta mais a soma de criticalidades relativas locais da ação em cada plano a qual pertence.

Uma outra questão abordada por esta proposta é o tempo de duração de uma ação. O tempo inicial da ação é medido usando um grafo ordenado topológico, considerando o tempo restante das ações antecedentes e irmãs. Com isto, foram propostos dois tipos de estratégias para atualizar a criticalidade dos agentes: uma estratégia é sensível ao tempo e a outra aos eventos.

O que as outras abordagens tinham de não considerar características do domínio da aplicação, esta tem em excesso e acaba sendo o seu maior problema, pois aumenta o acoplamento entre a forma que os agentes implementam seus planos e a forma que a proposta precisa que eles sejam implementados. Além de ser mais intrusiva quanto à implementação interna do agente, já que, em um sistema multiagentes aberto, não é possível identificar nem *a priori* nem *a posteriori* a implementação interna do agente, somente se ele está em conformidade com o sistema no qual está entrando segundo as suas interações, ou não.