6
O *Framework* Avestruz

O principal objetivo do Avestruz é gerar ferramentas para procurar e classificar, automaticamente, documentos presentes em diferentes locais indicados pelo usuário. As ferramentas geradas podem ser altamente ativas, no sentido de que o trabalho é todo automático: a princípio, a única entrada requerida são os locais onde os documentos estão.

Deve-se ter em mente que, sendo um framework, o Avestruz necessita ser instanciado segundo as necessidades dos seus usuários. Os hot spots existentes são: o tipo de documento analisado, o algoritmo de classificação, o tipo de relatório gerado na saída e o grau de inteligência do agente.

A flexibilidade no tipo de documento analisado é fundamental para que as ferramentas geradas sejam capazes não somente de processar os diversos formatos de documentos já existentes (como XML, HTML, Word for Windows, PostScript, PDF, etc), como também de se adaptar rapidamente a novos formatos; ou ainda, a dar suporte a formatos proprietários de uma organização.

A flexibilidade no algoritmo de classificação é essencial, pois cada aplicação pertencente ao domínio coberto pelo framework possui um requisito de precisão – algumas aplicações necessitam de uma classificação mais confiável e precisa do que outras.

A flexibilidade no relatório de saída também é importante, pois dependendo da aplicação, diferentes ações podem ser tomadas. Por exemplo, uma aplicação poderia simplesmente gerar um relatório para um gerente de projeto a fim de que ele saiba o que a sua equipe tem pesquisado, ao passo que uma outra pode fazer uso dessa informação para armazenar os documentos em uma base de dados comum de consulta.

O grau de inteligência do agente está diretamente relacionado às decisões que serão tomadas enquanto este realiza as buscas e classificações. Por exemplo, o agente pode ser capaz de "lembrar" de documentos já analisados, a fim de evitar dupla análise numa pesquisa (memória temporária) ou em várias pesquisas (memória persistente). Ou ainda, o agente pode ser capaz de buscar documentos além dos locais previamente indicados pelo usuário (com uma maior autonomia).

A arquitetura do framework teve como principal objetivo e desafio desacoplar inteiramente as questões referentes ao algoritmo de classificação utilizado das referentes à plataforma de busca (localização, seleção e gerenciamento de novos documentos), numa clara aplicação de engenharia de software e conceitos de separation of concerns [49]. Uma vez definidas as questões da plataforma de busca (tipo de documento analisado, o tipo de relatório gerado na saída e o grau de inteligência do agente), sobram as questões referentes ao algoritmo de classificação, que podem ser abordadas por um especialista nas características do domínio dos documentos categorizados. A grande vantagem é a possibilidade de testar diversos tipos de algoritmos, alguns deles inclusive oriundos de adaptações de algoritmos já existentes através de heurísticas segundo os tipos de documentos a serem analisados. Em outras palavras, uma pessoa especialista na área de algoritmos tem em mãos, com o framework, uma potente ferramenta de teste, uma espécie de laboratório onde poderia realizar suas experiências, preocupando-se somente com as questões referentes à classificação, e reaproveitando toda a infra-estrutura de plataforma já desenvolvida. É como se houvesse uma subdivisão nos hot spots do framework, pois se espera que o algoritmo de classificação seja muito mais alterado do que os demais hot spots do framework.

A estratégia utilizada para conseguir os objetivos de independência entre questões de plataforma e classificação foi concentrar os frozen spots na camada de gerenciamento do sistema multi-agentes (que lida com questões como cooperação e coordenação dos agentes), os hot spots de plataforma, na camada de gerenciamento e no próprio agente de software, e os hot spots de classificação, somente no agente, posicionando-o como um usuário de serviços externos. Desta forma, uma vez instanciada a plataforma, o instanciador do framework fica livre de preocupações referentes a questões de agentes de software, focando sua

atenção no algoritmo de classificação. Em primeira aproximação, ele nem precisaria saber que o sistema efetivamente é multi-agentes – este "detalhe" ficaria totalmente encapsulado.

É importante notar que o caráter ativo das ferramentas geradas pelo framework tem por objetivo minimizar a necessidade de interferência do usuário durante o processo de execução, evitando assim as já conhecidas falhas decorrentes da falta de paciência (para não dizer empenho) das pessoas. Outro fator importante é que as instanciações do framework não estão limitadas a aplicações completas - é possível gerar componentes independentes, prontos para serem acoplados a aplicações da área de gestão de conhecimento.

Para fins de prova de conceito, foram realizadas quatro instanciações do framework:

- Uma aplicação completa, batizada de *Webclipper*, para realização de *clipping* na Internet;
- Um componente (batizado de KM *Probe*), para ser acoplado em uma outra ferramenta de gestão de conhecimento, atualmente em desenvolvimento no laboratório TecComm;
- Uma aplicação completa, batizada de *Site Seeker*, para a realização de busca de palavras em páginas de *websites*;
- Um componente (batizado de *Semantic Probe*), para ilustrar como a *web semântica* pode contribuir para a área de classificação de documentos.