

Uso de *Wrappers* e *Web Services* para a Construção de Sistemas de Modelagem de Nicho Ecológico em SOA

Fabiana S. Santana^{1,2}, Jefferson C. Ferreira¹, Renata L. Stange¹, Antonio M. Saraiva¹

¹Escola Politécnica – Universidade de São Paulo (USP) – São Paulo – SP – Brazil

²Centro Universitário da FEI – São Bernardo do Campo – SP – Brazil

{fabiana.santana, rlstange, saraiva}@usp.br;
jefferson.ferreira@poli.usp.br

Abstract. *Ecological niche modelling combines environmental and species data for generating models to describe the probabilistic distribution of a species in a region. Monolithic software packages for ecological niche modelling are not enough to attend the problem requirements. However, the reuse of functions implemented in those packages enables the fast implementation of the modelling process. This paper describes wrappers and web services application in order to reuse legacy systems and propose a comprehensive solution for the problem. In addition to implement the ecological niche modelling process, the solution presented here may be extended to other relative domains, showing its technical potential.*

Resumo. *A modelagem de nicho ecológico combina dados ambientais e de espécies para descrever a distribuição probabilística da espécie em determinada região. Sistemas monolíticos para modelagem não são capazes de atender todos os requisitos do problema. Contudo, o reuso das funções implementadas nestes pacotes permite a rápida implementação de um sistema para modelagem. Este artigo descreve a aplicação de wrappers e de web services para o reuso dos sistemas legados e propõe uma solução abrangente para o problema. Além de implementar o processo de modelagem, esta solução pode ser estendida para outros domínios relacionados em ecologia, mostrando o seu potencial técnico.*

1. Introdução

O nicho ecológico de uma espécie [Phillips et al. 2006] é o conjunto de condições que permitem a sua sobrevivência desconsiderando fatores externos, como a influência do homem, as interações bióticas e as barreiras geográficas. Os pontos de ocorrência de uma espécie, em dados georreferenciados (latitude e longitude), correspondem aos locais onde a espécie foi encontrada. Os dados ambientais, como clima, altitude, solo e vegetação, devem ser organizados em camadas do espaço ecológico correspondente à região estudada, também na forma de dados georreferenciados. Algoritmos capazes de encontrar regiões similares às regiões onde as espécies foram de fato encontradas buscam relações não aleatórias entre os dados de ocorrência da espécie e os dados ambientais relevantes para sua sobrevivência [Anderson et al. 2003]. Eles se baseiam no conceito de nicho ecológico e na hipótese de que, se uma espécie pode ser encontrada

sob certas condições, então ela deve ser capaz de sobreviver e se reproduzir em qualquer lugar sob as mesmas condições. O resultado é um modelo que corresponde à função de distribuição da probabilidade de ocorrência da espécie nas coordenadas geográficas da região de estudo. Os modelos podem representar a distribuição das espécies no passado, no presente e no futuro, e normalmente são projetados em mapas da região geográfica estudada [Peterson 2001], em escala de cores, onde cada cor representa uma probabilidade específica. Portanto, a modelagem de nicho ecológico busca gerar modelos de distribuição probabilística de uma espécie em uma determinada região, a partir de dados ambientais e de dados de presença da espécie em estudo, que são essenciais para o estudo e o gerenciamento da biodiversidade.

Apesar de desconsiderar fatores externos ao ciclo de vida natural das espécies, os modelos de nicho já foram aplicados na solução de diversos problemas em biodiversidade. Por exemplo, já serviram de base para propor cenários de uso sustentável do meio-ambiente, avaliar o potencial de espécies invasoras, avaliar as mudanças climáticas e os seus impactos na biodiversidade e delinear o potencial de rotas de infecções e doenças [Santana et al. 2008].

Entre os principais desafios para a construção de uma solução de software abrangente para a modelagem de nicho ecológico estão: 1) Formalizar o processo de modelagem de nicho ecológico [Santana et al. 2008]; 2) Realizar o tratamento de metadados, com o uso de padrões já existentes, para prover interoperabilidade [Santana et al. 2007]; 3) Fazer a integração com SIG, Sistemas de Informação Geográfica, bancos de dados distribuídos e novos algoritmos de modelagem; 4) Evoluir a interface, considerando critérios de usabilidade e estabilidade; 5) Construir bases de dados para armazenar informações sobre espécies e sobre os modelos gerados; 6) Definir ferramentas e mecanismos de busca e importação de dados; 7) Realizar análises estatísticas mais avançadas para a seleção de modelos e integrá-las ao processo de modelagem; e 8) Incorporar técnicas e equipamentos para realizar a computação de alto desempenho. A solução deve ser desenvolvida a partir de uma iniciativa *open source*, utilizando ferramentas de software livre e padrões, para evitar restrições de uso e permitir o desenvolvimento colaborativo.

Já existem pacotes de software para a modelagem de nicho ecológico que implementam algumas destas funcionalidades. Os principais são o openModeller [Santana et al. 2006], o DesktopGarp [<http://nhm.ku.edu/desktopgarp>] e o MaxEnt [Philips et al. 2006]. O openModeller é o mais completo e permite analisar, visualizar e construir cenários ambientais no presente, no passado ou no futuro, além de implementar diversos algoritmos de modelagem. O DesktopGarp permite realizar múltiplas projeções a partir de um mesmo modelo, porém seu único algoritmo de modelagem é o Garp [Stockwell e Peters 1999]. O MaxEnt não oferece facilidades adicionais, porém o seu único algoritmo, também chamado de MaxEnt [Philips et al. 2006], é um dos mais utilizados atualmente por sua alegada acurácia. Os três pacotes de software possuem interfaces *desktop* e/ou console. Eles foram desenvolvidos como sistemas monolíticos e não seguem padrões de implementação, resultando em restrições arquiteturais para atender todos os requisitos da modelagem de nicho ecológico, especialmente os relativos à integração e interoperabilidade de sistemas. O usuário normalmente utiliza mais de um pacote de software diferente para gerar um modelo e fazê-lo não é simples.

Em Santana et al. (2007), uma infraestrutura orientada a serviços para a modelagem de nicho ecológico foi proposta. Esta solução foi desenvolvida de acordo com os padrões J2EE [Stal 2006] para SOA, *Service-Oriented Architecture*, conforme apresentado em Stange et al. (2008), e permite a integração e a interoperabilidade de sistemas usando *web services* e XML. No entanto, o conjunto de funcionalidades implementado pelo openModeller, pelo DesktopGarp e pelo MaxEnt é considerável, principalmente no que diz respeito aos algoritmos de modelagem de nicho ecológico. Portanto, é interessante propor uma forma de se utilizar, na infraestrutura, os recursos implementados por estes pacotes, na forma de serviços.

Este trabalho apresenta uma proposta para construir serviços para modelagem de nicho ecológico a partir de sistemas legados, utilizando *wrappers* e *web services*, e mostra como integrá-los para construir uma aplicação capaz de executar o processo de modelagem na infraestrutura de serviços. Para a implementação, a biblioteca do openModeller foi considerada como sistema legado, mas a mesma solução pode ser aplicada aos demais pacotes de software, com pequenas diferenças na implementação.

Desta forma, é possível encapsular e disponibilizar, em uma solução orientada a serviços, as funcionalidades dos pacotes de software existentes para modelagem de nicho ecológico. A vantagem desta abordagem é que, ao contrário destes pacotes, a infraestrutura orientada a serviços dispõe de recursos para integração com diversas tecnologias e foi projetada para atender a todos os requisitos identificados para o problema. Vários dos requisitos de modelagem já estão integrados, a saber: serviços de SIG, serviços para persistência em bancos de dados distribuídos, serviços para modelagem providos por algoritmos paralelos em um *cluster*, algoritmos adaptativos e serviços de importação de dados.

2. Material e Métodos

Esta seção apresenta a infraestrutura orientada a serviços para a modelagem de nicho ecológico, o conceito de *wrapper* e a tecnologia para a implementação de serviços (*web services*).

2.1. Infraestrutura orientada a serviços para a modelagem de nicho ecológico

A infraestrutura orientada a serviços para a modelagem de nicho ecológico apresentada em Santana et al. (2007) pode ser resumida pela Figura 1.

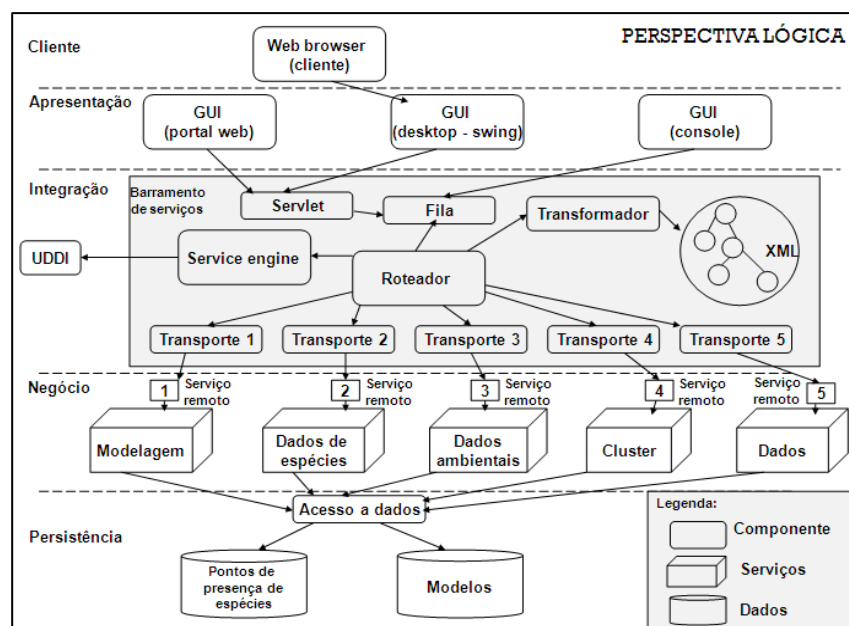


Figura 1 – Infraestrutura para a modelagem de nicho ecológico.

A camada cliente contém os dispositivos de acesso às aplicações. A camada de apresentação disponibiliza diversos tipos de interfaces do usuário, através de um portal web ou outros tipos de interface, incluindo opções para *desktop* e console. A camada de integração é responsável pela integração dos serviços internos ou externos, necessários para realizar um processo de negócio completo, através de um barramento de serviços (parte escura da Figura 1), que faz o gerenciamento das requisições de serviços através do roteador. O caminho de um serviço começa por sua solicitação e posterior inclusão na fila, o que é feito a partir das camadas cliente, usando um *web browser*, ou de apresentação, usando interface *desktop* ou console. O barramento prevê um *service engine* para recuperação e invocação de serviços dinamicamente. A camada de negócio contém os serviços de modelagem e os demais serviços necessários para o seu processamento. A camada de persistência oferece recursos de banco de dados, para armazenar modelos e pontos de presença das espécies.

2.2 Uso de Wrappers

A solução encontrada para encapsular as diversas funcionalidades já implementadas nos pacotes de software para a modelagem de nicho ecológico foi o uso de wrappers. Esta solução permite o desacoplamento das funcionalidades da modelagem de nicho ecológico das tecnologias e ferramentas utilizadas em sua implementação nos pacotes de software disponíveis. A solução a ser disponibilizada vai utilizar os serviços desacoplados e orquestrados, como no portal de modelagem de nicho ecológico. A Figura 2 mostra como os *wrappers* são utilizados para transformar o sistema legado em serviços. Inicialmente, deve ser feita a extração das classes que devem compor o serviço. Estas classes devem ser ajustadas para permitir a criação dos *wrappers*. Cada conjunto de classes deve corresponder a um componente. Este componente será, posteriormente, disponibilizado na forma de um serviço.

A implementação das classes *wrappers* [Doucet et al. 2001; Juric et al. 2000; Thiran e Hainaut 2001] a partir de um sistema monolítico deve ter o objetivo de minimizar as alterações no código do sistema original. O excesso de alterações pode exigir esforços demasiados para a manutenção do sistema e aumentar a complexibilidade dos testes funcionais. O primeiro passo para se implementar um *wrapper* é definir qual a parte do sistema legado que

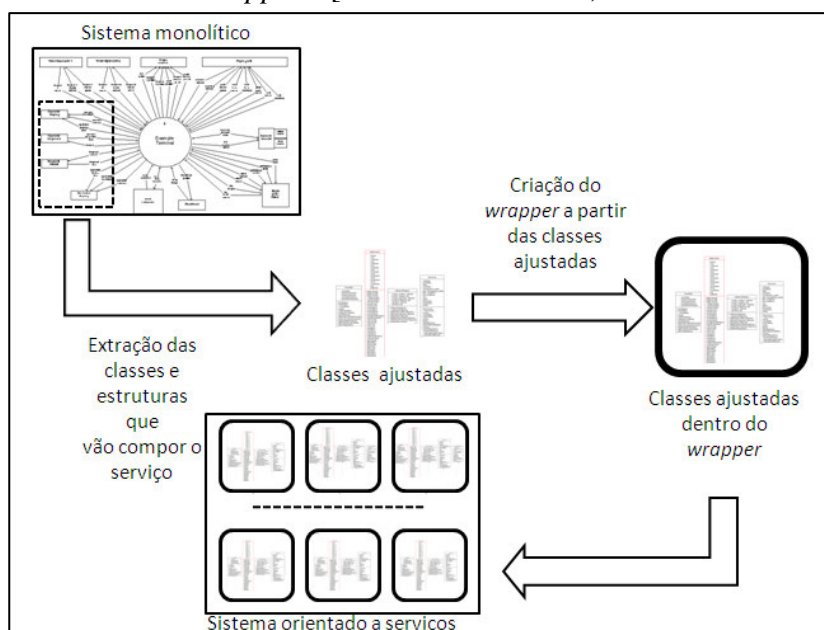


Figura 2 – Uso de wrappers.

será encapsulada por ele. O *wrapper* deve encapsular um trecho de código com um escopo funcional bem definido, para permitir o reuso do software e simplificar o entendimento do sistema orientado a serviços.

2.3. Tecnologia para a Implementação de Serviços

Aplicações com características distribuídas e heterogêneas requerem uma abordagem de computação distribuída para a integração [STAL, 2002]. A tecnologia de *web services* é uma tendência em SOC, *Service Oriented Computing*, para reduzir o acoplamento entre cliente e provedor de serviços em sistemas que utilizam a Internet como meio de comunicação. Além disso, padrões abertos tais como SOAP, *Simple Object Access Protocol*, para transmissão dos dados, e WSDL, *Web Services Description Language*, para definição dos serviços, facilitam a integração de componentes na *web*.

Um *web service* é baseado em XML e o transporte de dados ocorre de forma transparente em qualquer plataforma baseada em protocolos de serviços *web*. Segundo Endrei et al. (2004), as principais características de um *web service* são a independência de plataforma, a descrição bem definida de sua função, a disponibilidade para ser publicado e descoberto, e a capacidade de ser composto em uma aplicação distribuída. A composição de serviços permite realizar processos de negócios completos, e não somente simplificar a troca de informações.

A partir de um conjunto de serviços que cooperam entre si para solucionar um determinado problema, um mecanismo para controlar a execução desses serviços na forma de um processo é necessário. Esse mecanismo pode ser implementado utilizando um barramento de serviços e uma linguagem específica, como BPEL, *Business Process Specification Language*. O barramento de serviços permite a integração de serviços e a composição de novas aplicações. Na infraestrutura orientada a serviços para a modelagem de nicho ecológico foi adotado o Open ESB [<https://open-esb.dev.java.net>] como solução de integração e a linguagem BPEL para descrever o processo e a interação com os *web services*.

3. Implementação de Serviços para Modelagem de Nicho Ecológico usando Wrappers Criados com a Biblioteca do openModeller

Esta seção apresenta os componentes de negócio identificados para a modelagem de nicho ecológico e, considerando a biblioteca do openModeller como o sistema legado, descreve a implementação dos *wrappers*. Em seguida, a criação de um serviço a partir do *wrapper* criado é apresentada, bem como a sua integração com a plataforma de serviços para modelagem de nicho ecológico.

3.1. Identificação dos Componentes de Negócio

Os componentes da modelagem de nicho ecológico devem corresponder aos requisitos do problema. Dentro da infraestrutura de serviços proposta em Santana et al. (2007), são necessários os seguintes componentes, descritos no diagram de componentes apresentado na Figura 3: 1) Provedor de dados ambientais; 2) Provedor de dados de espécies; 3) Provedor de algoritmos de modelagem; 4) Geradores de mapas e imagens; 5) Provedor de persistência de dados; 6) Processos coordenadores; e 7) Portal. O portal recebe as requisições e as encaminha ao coordenador do provedor de serviços de

modelagem. O provedor de serviços de modelagem acessa o coordenador do processo, que distribui as tarefas entre os demais componentes. Os provedores de dados de espécies e de dados ambientais permitem obter as entradas necessárias para executar a modelagem, o que é feito usando os algoritmos específicos. Uma vez gerados os resultados, o provedor de mapas e imagens é acessado para gerar os mapas que representam os modelos de nicho ecológico. O provedor de persistência permite armazenar os modelos gerados.

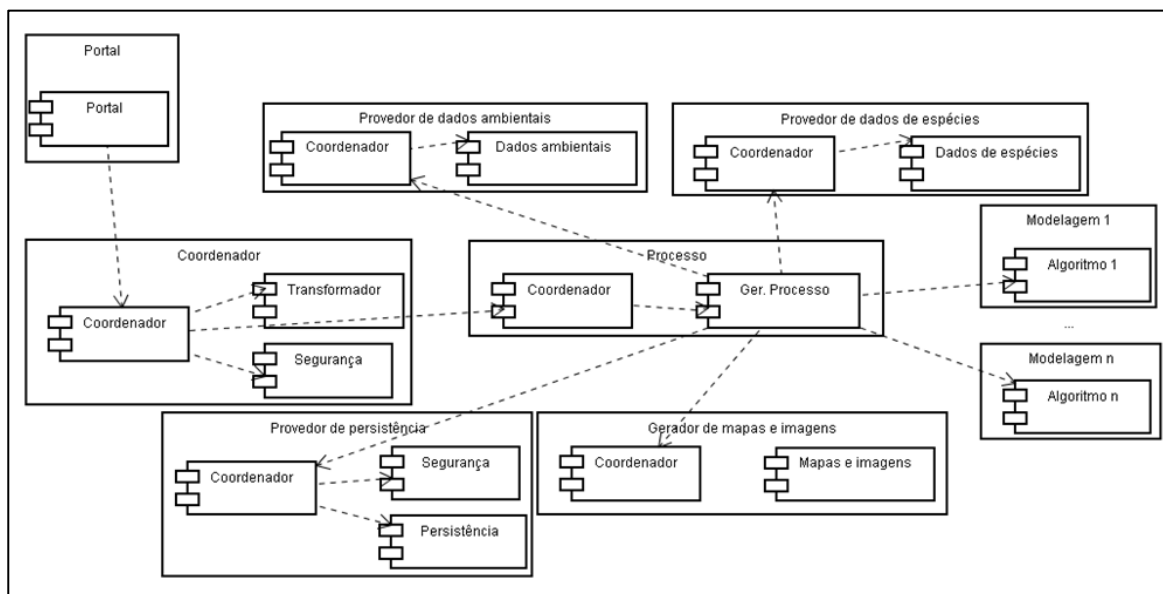


Figura 3 – Diagrama de componentes para a modelagem de nicho ecológico.

Os serviços devem ser implementados de acordo com os componentes identificados, e alocados de forma a manter o desacoplamento, a fim de que seja possível, por exemplo, obter os dados ambientais e de espécies a partir de provedores diferentes, utilizar algoritmos de modelagem supridos por diferentes provedores (e.g.: cluster e provedor openModeller, a ser criado), orquestrar as aplicações e disponibilizá-las em um portal de serviços de modelagem de nicho ecológico.

A implementação dos serviços já disponíveis nos sistemas legados, a partir do diagrama de componentes apresentado, deve ser feita usando *wrappers*.

3.2. Implementação de Classes *Wrappers* usando a Biblioteca do openModeller

A biblioteca do openModeller, disponível em <http://openmodeller.sourceforge.net/>, implementa algumas funções necessárias para a modelagem de nicho ecológico e será considerada como sistema legado para exemplificar a implementação do *wrapper*.

Para descrever a criação de um *wrapper*, foi selecionada a projeção de mapas a partir de um modelo gerado, correspondente ao componente gerador de mapas e imagens do diagrama de componentes para a modelagem de nicho ecológico. Como a biblioteca do openModeller foi desenvolvida em linguagem C++, a implementação do *wrapper* será feita na mesma linguagem, embora esta não seja uma restrição da solução.

Para implementar um *wrapper*, o primeiro passo é separar a parte do código que se quer encapsular. É preciso identificar o subprograma, ou a classe, a partir do qual a

funcionalidade correspondente ao componente é chamada. No caso do *wrapper* de projeção de mapas do openModeller, esta função é a própria função *main* e encontra-se no arquivo *om_console.cpp*. Uma vez identificado este subprograma, ou classe, é necessário identificar todos os demais subprogramas, ou classes, a ele relacionados. No caso do openModeller, as demais classes a serem encapsuladas no *wrapper* estão no arquivo *request_file.cpp*. Em seguida, deve-se separar o código que deve ser encapsulado, por exemplo através da criação de uma nova classe para contê-lo. No caso da projeção de mapas, foi criada a classe *OmProjection*. Deve-se notar que os ajustes podem se estender a várias partes do sistema, dependendo da organização, ou falta dela, do sistema legado. Como a biblioteca do openModeller está organizada, o trabalho para implementar o *wrapper* de projeção foi relativamente simples.

Uma vez que o código a ser encapsulado pelo *wrapper* esteja identificado e separado, deve-se proceder à implementação do próprio *wrapper*, usando a classe ou estrutura criada anteriormente como base para a sua construção.

```
class OmProjectionWrapper {
private:
    // Ponteiro para a classe encapsulada
    OmProjection * om_projection;
public:
    // Construtor da classe wrapper instancia construtor da classe
encapsulada
    OmProjectionWrapper (char *request_file) {
        om_projection = new OmProjection ( request_file );
    }
    // Destrutor da classe wrapper remove memória alocada
    ~OmProjectionWrapper ( ){
        delete om_projection;
    }
};
```

Figura 4 – Implementação do wrapper correspondente à projeção de um modelo.

Um wrapper em C++ é composto por uma classe, que contém ponteiros para cada uma das classes que foram encapsuladas na etapa anterior. Os métodos da classe wrapper, na maior parte dos casos, resumem-se a chamar logicamente os métodos das outras classes. Para facilitar o entendimento, a construção de um wrapper pode ser detalhada pelo seguinte roteiro: 1) Criar uma classe *wrapper* que contem um ou mais ponteiros para as classes encapsuladas na etapa anterior; 2) Incluir no construtor da classe *wrapper* as chamadas para os construtores das demais classes; 3) Incluir no destrutor da classe *wrapper* as chamadas para os construtores das demais classes e a liberação de espaços de memória que eventualmente tenham sido alocados pelo construtor; e 4) Verificar os casos em que ocorre a sobrecargas de operadores nas classes encapsuladas, e redefini-las quando necessário. A Figura 4 mostra a implementação do wrapper correspondente à projeção, utilizando a biblioteca do openModeller, em detalhes. Na figura, os métodos da classe *wrapper* são compostos por chamadas aos métodos da classe encapsulada através das referências declaradas na classe *wrapper*.

3.3. Integração e Composição de Serviços Desenvolvidos a partir de *Wrappers*

A definição de um serviço envolve a especificação de uma interface, descrevendo as entradas e as saídas do serviço, e as interações que o serviço pode manter com outros serviços.

Considerando que as funcionalidades de um sistema monolítico podem ser encapsuladas por *wrappers*, as classes que as representam devem ser implementadas como serviços. Para fazê-lo utilizando a solução implementada na infraestrutura de serviços para a modelagem de nicho ecológico, é necessário que seja criado um *web service* para cada classe *wrapper*, contendo a descrição da interface contida na aplicação. Nesta solução, cada *web service* expõe sua interface no barramento de serviços a partir da sua descrição em WSDL e a orquestração desses serviços é feita utilizando BPEL.

A Figura 5 mostra o esquema do processo BPEL. O processo BPEL é iniciado (*Start*) e uma requisição do cliente (*receive*) é procesada. Serviços externos são invocados para execução de procedimentos (*invoke*) e, finalmente, uma resposta é retornada ao cliente (*reply*). Um processo BPEL pode interagir com outros processos em BPEL ou com outros *web services* [Salter et al. 2008].

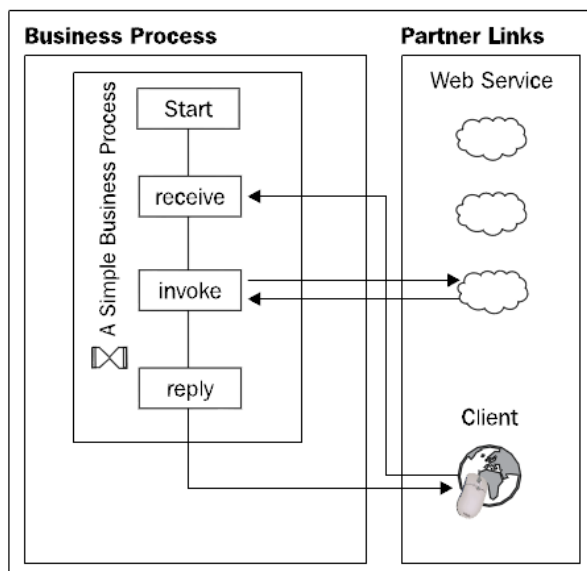


Figura 5 – Processo em BPEL [Salter et.al. 2008].

O processo é especificado em um arquivo XML e o seu *deploy* em BPEL é feito em no barramento de serviços.

4. Implementação da Modelagem de Nicho Ecológico usando *Wrappers* e *Web Services*

Para se construir um sistema para a modelagem de nicho ecológico utilizando os *wrappers* e os conceitos a respeito de *web services* discutidos anteriormente, foi preciso conhecer o processo de modelagem, definido em Santana et al. (2008).

As etapas deste processo estão especificadas na Figura 6. O processo consiste em uma etapa inicial em que o problema é definido, os pontos de espécie e os dados ambientais necessários para a geração do modelo são obtidos, é realizada a pré-análise para verificar se o modelo pode ser gerado, os algoritmos e os correspondentes parâmetros são selecionados. Em seguida, o modelo é gerado e projetado e, na etapa

final, o modelo deve ser analisado e validado pelo pesquisador em modelagem de nicho ecológico.

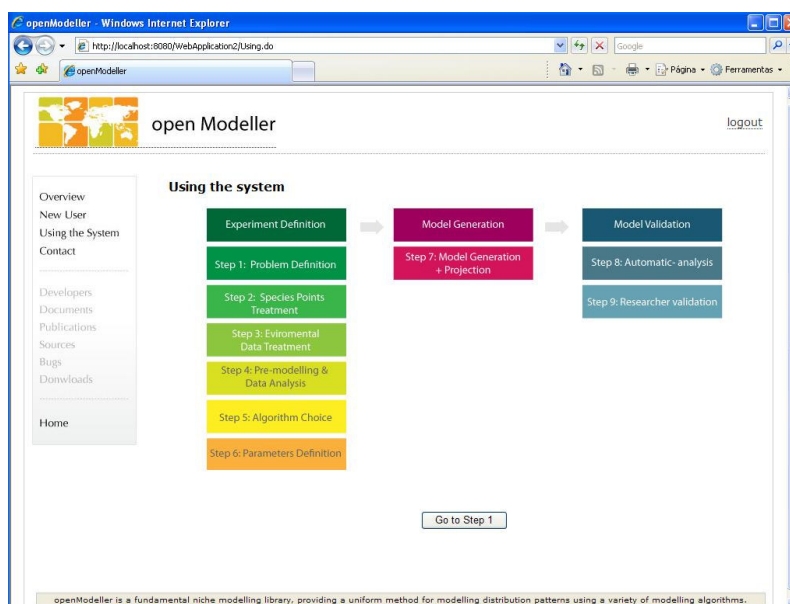


Figura 6 – O processo de modelagem de nicho ecológico.

A partir da implementação dos *wrappers* correspondentes aos componentes identificados e do processo de modelagem na infraestrutura de serviços para a modelagem de nicho ecológico, foi possível construir uma aplicação para realizar a modelagem via *web*. Esta aplicação está disponível a partir de www.pcs.usp.br/~laa/.

Para exemplificar o uso do *wrapper* de projeção *OmProjectionWrapper* e do correspondente *web service*, a Figura 7 mostra o resultado da projeção de um modelo que contém a distribuição mundial de legumes, conforme estudo realizado em Yesson et al. (2007).

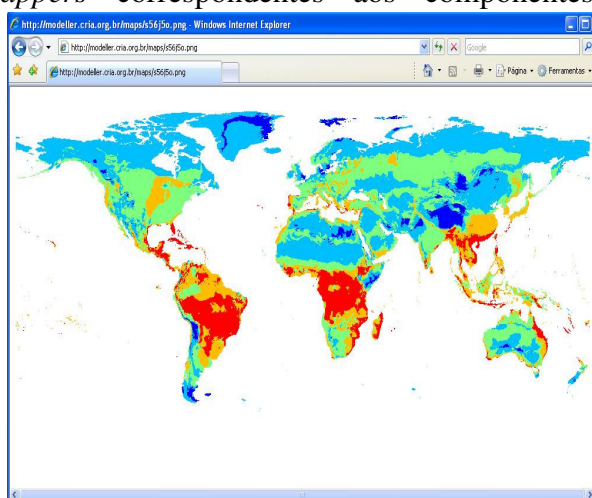


Figura 7 – Projeção da distribuição mundial de legumes.

5. Conclusão

A solução apresentada neste trabalho mostrou-se viável e permitiu, de forma simples, incorporar todas as funcionalidades disponíveis no openModeller ao processo de modelagem de nicho ecológico na infraestrutura de serviços.

Com base na mesma proposta, é possível agregar à modelagem de nicho ecológico todas as funcionalidades implementadas pelos demais pacotes de software para modelagem de nicho ecológico. Desta forma, os pesquisadores em biodiversidade podem contar com uma solução única, abrangente e integrada, para estudar a distribuição das espécies através da modelagem de nicho ecológico, uma vez que esta solução

elimina a necessidade de conhecer e lidar com pacotes de software diferentes e complexos.

Por ser completamente baseada em padrões, a infraestrutura de serviços e a solução apresentada neste trabalho podem ser estendidas para outras áreas do conhecimento que apresentem requisitos de software semelhantes ao deste problema.

Referências

- Anderson, R. P., Lew, D., Peterson, A.T. (2003) “Evaluating predictive models of species' distributions: criteria for selecting optimal models”. *Ecological Modelling*. V. 162, p. 211-232.
- Doucet, F. Gupta, R. Otsuka, M. Schaumont, P. Shukla, S. (2001) “Interoperability as a design issue in C++ based modeling environments”. *Proc. System Synthesis*.
- Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M., Newling, T. (2004) “Patterns – SOA and Web Services”. IBM Redbook.
- Juric, M. B., Rozman, I., Hericko, M., Domajnko, T. (2000) “Integrating legacy systems in distributed object architecture”. *ACM SIGSOFT Software Engineering Notes*. V. 25-2, pp. 35-39.
- Peterson, A. T. (2001) “Predicting species geographic distributions based on ecological niche modeling”. *CONDOR*. V. 103, p. 599-605.
- Phillips, S. J., Anderson, R. P., Schapire, R. E. (2006) “Maximum entropy modelling of species geographic distributions”. *Ecological Modelling*. V. 190, p. 231-259.
- Salter, D., Jennings, K. (2008) *Building SOA-Based Composite Applications Using NetBeans IDE 6*. Packt Publishing Ltd., Birmingham, UK, 287 pp.
- Santana, F. S., Fonseca, R. R., Saraiva, A. M. et al. (2006) “openModeller - an open framework for ecological niche modeling: analysis and future improvements”. *WCCA*. P. 787-792.
- Santana, F. S., Murakami, E., Saraiva, A. M. et al. (2007) “Uma arquitetura de referência para sistemas de informação para modelagem de nicho ecológico”. *Anais do 6º Congresso Brasileiro de Agroinformática – SBIAgro*. P. 101-105.
- Santana, F. S., Siqueira, M. F., Saraiva, A. M., Corrêa, P. L. P. (2008) “A reference business process for ecological niche modelling”. *Ecological Informatics Journal*, Elsevier. V. 3-1, p.75-86.
- Stal, M. (2002) “Web Services: Beyond Component-Based Computing”. *Communications Of The ACM*. Vol. 45, No. 10, 71-76.
- Stange, R. L., Santana, F. S., Saraiva, A. M. (2008) “Applying J2EE patterns to develop a SOA-based architecture for ecological niche modeling”. *Proceedings of the 7th International Information and Telecommunication Technologies Symposium*.
- Thiran, P., Hainaut, J. L. (2001) “Wrapper development for legacy data reuse”. *Reverse Engineering*, 2001. Pp. 198-207.
- Yesson, C., Brewer, P. W., Sutton, T. et al. (2007) “How Global Is the Global Biodiversity Information Facility?”. *PLoS ONE* 2(11): e1124.