



Gustavo de Sá Carvalho Honorato

NCL-Inspector

Uma ferramenta para inspeção de aplicações NCL

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC-Rio

Orientador: Prof. Simone Diniz Junqueira Barbosa

Rio de Janeiro
Março de 2010



Gustavo de Sá Carvalho Honorato

NCL-Inspector

Uma ferramenta para inspeção de aplicações NCL

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Simone Diniz Junqueira Barbosa
Orientador
Departamento de Informática — PUC-Rio

Prof. Luiz Fernando Gomes Soares
Departamento de Informática – PUC-Rio

Prof. Edward Hermann Haeusler
Departamento de Informática – PUC-Rio

Prof. José Eugênio Leal
Coordenador Setorial do Centro Técnico Científico — PUC-Rio

Rio de Janeiro, 25 de Março de 2010

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Gustavo de Sá Carvalho Honorato

Graduado em Ciência da Computação pela Universidade Federal Fluminense (UFF) em 2006. Atualmente, integra o *Semiotic Engineering Research Group*, desenvolvendo pesquisa na área de TV Digital e Interação Humano-Computador.

Ficha Catalográfica

Honorato, G. S. C.

NCL-Inspector / Gustavo de Sá Carvalho Honorato; orientador: Simone Diniz Junqueira Barbosa. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2010.

v., 120 f: il. ; 29,7 cm

1. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Tese. 2. ferramenta de inspeção para NCL. 3. ferramenta de análise estática de código. 4. NCL. 5. desenvolvimento hipermídia. I. Barbosa, S. D. J.. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Gostaria de agradecer a Simone, pois sem sua contribuição este trabalho não teria sido possível. Agradeço por todos os ensinamentos, por ter me acompanhado nos primeiros passos em IHC e nas descobertas sobre TV Digital; também pelas discussões que tanto contribuíram com esse trabalho.

Aos professores do Departamento de Informática que contribuíram para a minha formação.

Aos professores da UFF que despertaram em mim o interesse pela pesquisa e pelo mestrado.

Aos colegas do Laboratório SERG, onde encontrei um espaço agradável para desenvolvimento de meu trabalho.

Aos colegas do Laboratório Telemídia e em especial ao professor Carlos Salles (Carlão), que possibilitaram a execução dos experimentos com o NCL-Inspector.

De forma especial, à Leandra. Pela maneira única que me ama e me compreende dia a dia. Por sua paciência e incentivo em cada momento da minha vida.

Também ao meu irmão, Rafael, pessoa de extrema importância na minha vida.

Ao amigo Hildebrando (Will) pela participação na minha vida pessoal, acadêmica e profissional desde quando nem imaginávamos realizar o mestrado.

E enfim, à CAPES e à PUC-Rio que possibilitaram minha atuação no contexto da pesquisa através do apoio financeiro.

Resumo

Honorato, G. S. C.; Barbosa, S. D. J.. **NCL-Inspector. Uma ferramenta para inspeção de aplicações NCL**. Rio de Janeiro, 2010. 120p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Ginga-NCL e a linguagem NCL foram recomendadas como ambiente de multimídia interativo e linguagem para IPTV, pela União Internacional das Telecomunicações (ITU – do Inglês: *International Telecommunication Union*). Para promover o uso dessas tecnologias, é necessária a criação de ferramentas que ajudem a desenvolver aplicações usando NCL. Atualmente, o suporte de ferramentas para o desenvolvimento em NCL é um pouco limitado. Existem apenas poucos sistemas que auxiliam o desenvolvimento de aplicações NCL. Esses programas não ajudam muito os desenvolvedores em avaliar se o código NCL está bem escrito. Neste trabalho apresentamos o NCL-Inspector, um sistema de críticas para código NCL, que objetiva aumentar a capacidade do desenvolvedor em detectar aplicações NCL suscetíveis a erros. Também foram levantados requisitos desejáveis em sistemas de críticas para código. A forma como esses requisitos foram apresentados possibilita que sejam aplicados em qualquer sistema de crítica de código fonte.

Palavras-chave

ferramenta de inspeção para NCL; ferramenta de análise estática de código; NCL; desenvolvimento hipermídia;

Abstract

Honorato, G. S. C.; Barbosa, S. D. J.. **NCL-Inspector. A NCL code inspection tool**. Rio de Janeiro, 2010. 120p. MsC Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Ginga-NCL and the NCL language were selected as a recommendation as the interactive multimedia environment and language for IPTV by the International Telecommunication Union (ITU). In order to promote the use of these technologies, it is necessary to create tools to help developing applications using NCL. At present, the support provided by tools for NCL development is quite limited. There are only few available systems that can create NCL applications. These softwares do not help developers in assessing the quality of the written NCL code. In this dissertation, we propose NCL-Inspector, a critique system of the NCL code, which aims at leveraging the developer's skills in detecting error-prone NCL applications. Also, we specified the requirements for critiquing systems for source code inspection.

Keywords

NCL inspection tool; static code analysis tool; NCL; hypermedia development;

Sumário

1	Introdução	11
1.1	Motivação	11
1.2	Objetivos	12
1.3	Estrutura da dissertação	13
2	Fundamentação	14
2.1	Problemas de qualidade de código	14
2.2	Estendendo XML Schemas	16
3	Trabalhos Relacionados	21
3.1	Eclipse Java Development Tools (JDT)	22
3.2	PMD	22
3.3	CheckStyle	24
3.4	FindBugs	25
3.5	NCL-Validator	27
3.6	Considerações	27
4	O Sistema de Críticas	29
4.1	Requisitos do Sistema	29
4.2	Especificação	33
4.3	Adicionando novas regras ao NCL-Inspector	46
4.4	Usando o NCL-Inspector como uma biblioteca	61
5	Avaliação do Sistema de Críticas	64
5.1	Descrição do estudo	64
5.2	Resultados	69
6	Conclusões e trabalhos futuros	77
6.1	Trabalhos futuros	78
	Referências Bibliográficas	86
A	Compilando o NCL-Inspector a partir do código fonte	89
B	Preparando o ambiente de desenvolvimento de novas regras	92
B.1	Desenvolvendo novas regras	92
B.2	Testando as regras	97
B.3	Fazendo a implantação de novas regras	98
C	Questionário Pré-Avaliação do NCL-Inspector	99
D	Exemplo da Descrição do Problema usado no Experimento	102
D.1	Descrição	102
D.2	Exemplos de elementos media que violam essa restrição	102

E	Ficha de Apoio à Observação de Uso	103
F	Roteiro da Entrevista Pós-Avaliação do NCL-Inspector	105
F.1	Por que optou por utilizar a tecnologia escolhida (XSL ou Java)?	105
F.2	O que achou do mecanismo de regras?	105
F.3	O que você modificaria no processo de criação de regras?	105
F.4	Compare o esforço de elaboração de uma regra com o de inspeção manual dos programas NCL?	105
F.5	Você teria alguma sugestão de caráter geral?	105
G	Problemas encontrados no XML Schema do NCL	107
H	Catálogo de problemas de código NCL	108
H.1	Módulo Structure Functionality	108
H.2	Módulo Layout functionality	108
H.3	Módulo Components functionality	109
H.4	Módulo Interfaces Functionality	111
H.5	Módulo Presentation Specification Functionality	112
H.6	Módulo Linking Functionality	113
H.7	Módulo Connectors Functionality	114
H.8	Módulo Presentation Control Functionality	117
H.9	Módulo Timing Functionality	118
H.10	Módulo Reuse Functionality	118
H.11	Módulo Navigation Key Functionality	120
H.12	Módulo Animation Key Functionality	120

Lista de figuras

3.1	Perspectiva adicionada pelo JDT	22
3.2	Crítica proveniente do compilador, informando que uma variável declarada não está sendo usada.	23
3.3	Ambiente de desenvolvimento de regras XPath do PMD	23
3.4	Exemplo de sinalização de erros do PMD	24
3.5	Exemplo do CheckStyle em uso	25
3.6	Correção automática do CheckStyle	25
3.7	Ambiente de criação de regras do CheckStyle	26
3.8	Integração do FindBugs com o Eclipse	26
3.9	NCL-Validator sendo usado dentro do Composer.	27
4.1	Esboço simplificado da arquitetura do NCL-Inspector	34
4.2	Diagrama simplificado do processo de inspeção do NCL-Inspector	35
4.3	Interfaces que representam os elementos media e property da NCL.	36
4.4	Classes e interfaces principais do Mecanismo de regras do NCL-Inspector e seus relacionamentos.	38
4.5	Diagrama de objetos do mecanismo de regras	39
B.1	Página de download do Eclipse	93
B.2	Janela de instalação de plugins do Eclipse	94
B.3	Adicionando o site de instalação do Plugin do Maven.	94
B.4	Seleção dos plugins da integração com o Maven que serão instalados.	95
B.5	Criando um novo projeto Maven	95
B.6	Tela de configuração do projeto Maven	96
B.7	Gerando um pacote jar para implantação das regras.	98

Lista de tabelas

5.1	Comparação resumida do perfil de cada participante	69
6.1	Resumo dos requisitos implementados	78
G.1	Problemas encontrados no XML Schema da norma com data de 14/04/2009	107