

Análise Estrutural e Comportamental de Aplicações Ginga-NCL Orientada a Conformidade

Thiago M. Prota, Douglas Vêras, Carlos A. G. Ferraz
Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Recife, Brasil
{tmp, dvs,cagf}@cin.ufpe.br

ABSTRACT

This work aims to present a thesis proposal in the context of Ginga-NCL, proposes a solution that allows analyze the interactive TV applications in this context from two perspectives: structural and behavioral. Thus, seeks to present the theoretical context, emphasizing the problem identification and the proposed objectives. Moreover, presents research planning and development, detailing the current stage of work, the proposed solution and a proposed evaluation. Finally, details the state of the art, assessing the related works.

Categories and Subject Descriptors

I.7.2 [Document Preparation]: Hypertext/hypermedia; D.2.6 [Programming Environments]: Integrated environments

General Terms

Verification, Languages, Design e Standardization

Keywords

Compliance Analysis, Structural, Behavioral, NCL, Verification

1. CONTEXTO TEÓRICO

O Sistema Brasileiro de TV Digital (SBTVD) publicou em 2007 um conjunto de normas ABNT¹ que especificam todas as regras referentes à codificação, transmissão, modulação, difusão e recepção digital [11]. Dentre essas normas, uma merece destaque, a ABNT NBR 15606-2 [1], por definir o ambiente Ginga-NCL, que fornece suporte à construção de aplicações declarativas [13]. Mais recentemente, a ITU² (*International Telecommunication Union*) certificou o Ginga-NCL, juntamente com a linguagem NCL³, como a primeira recomendação internacional para suporte à interação e multimídia para dispositivos de IPTV (*Internet Protocol Television*) [14].

2. IDENTIFICAÇÃO DO PROBLEMA

Dado que o sistema de TV Digital é essencialmente um sistema distribuído, fica impraticável identificar a origem dos problemas sinalizados pelos usuários (junto ao SAC – Serviço de Atendimento ao Consumidor – dos grandes fabricantes de TV e também das emissoras) [11] quando não há um processo de avaliação de conformidade em cada componente deste sistema. Pois, para o funcionamento correto de um sistema distribuído,

além de ser essencial definir protocolos (normas) é imprescindível atestar o atendimento desses, para que cada componente, bem como o sistema, seja classificado como confiável [11] [12].

As aplicações desenvolvidas são componentes fundamentais para o sistema de TV Digital, pois sua execução pode apresentar problemas para o usuário, mesmo que toda a infraestrutura de transmissão e recepção esteja conforme as normas. [11] sugere que também é importante analisar a conformidade das aplicações, pois elas devem seguir normas precisas para o seu correto funcionamento. Segundo [10], as aplicações desenvolvidas para o Ginga-NCL podem apresentar três tipos de problemas: **Problemas sintáticos**: Quando não seguem a gramática da linguagem; **Problemas estruturais**: Quando, apesar de se limitar à gramática da linguagem, não respeitam a semântica estática da linguagem (ex.: referência e escopo); e **Problemas comportamentais**: Quando, apesar de seguir fielmente as normas, a computação induzida pela aplicação resulta em comportamentos inesperados (interações ou exibições indesejadas).

3. OBJETIVOS

Como objetivo geral propõe-se definir uma solução que busque evidenciar os problemas durante a fase de desenvolvimento das aplicações do Ginga-NCL. Para isto utiliza a abordagem orientada a conformidade, pois além de apontar o problema existente é importante fornecer a rastreabilidade até os trechos normativos [1] que fundamentaram a sua avaliação [11]. Isto é importante pois as normas para este contexto não apresentam as regras de forma estruturada, podendo definir uma regra em várias sessões ou até em normas diferentes. Além disto, dentre os principais desafios da pesquisa, este trabalho definiu como metas: criar uma solução completa (de acordo com as regras normativas) e confiável, que atestem a qualidade das aplicações; e apresentar os resultados de forma rápida e enxuta, a fim de minimizar o tempo e o esforço de análise. Para a obtenção do objetivo geral, foram definidos os seguintes objetivos específicos:

- Definir uma abordagem para estruturar as normas, afim de possibilitar rastreabilidade da avaliação da solução, além de permitir analisar a completude da solução;
- Definir um processo de testes formal para a ferramenta, afim de acompanhar o grau de confiabilidade da solução;
- Integrar a solução à plataforma Eclipse⁴ para permitir a sua utilização na definição de um ambiente de desenvolvimento para contexto Ginga-NCL; e

¹ www.abnt.org.br

² www.itu.int

³ www.ncl.org.br

⁴ www.eclipse.org

- Atestar a qualidade da ferramenta através de experimentos formais, principalmente no âmbito de desempenho, usabilidade e eficiência (em identificar e corrigir problemas).

4. Planejamento

A pesquisa deve consistir das seguintes etapas:

- I. Levantamento do estado da arte, através de um Mapeamento sistemático [6];
- II. Analisar o problema, os desafios e as metas, identificar e analisar os potenciais problemas e desafios a fim de formalizar as metas da pesquisa;
- III. Definir a proposta de solução, analisar os resultados das etapas anteriores, para identificar os requisitos e projetar uma solução de qualidade aderente;
- IV. Propor as avaliações, definir quais avaliações são relevantes para o contexto do protótipo e da solução;
- V. Desenvolver e avaliar o protótipo, definir, implementar e testar o escopo da solução a ser desenvolvido como prova de conceito. Além de, executar as avaliações definidas em IV a fim de encontrar oportunidades de melhoria no protótipo; e
- VI. Desenvolver e avaliar a solução final, evoluir o protótipo implementando e testando o restante do escopo, e realizando as melhorias identificadas em V. Além de, executar avaliações definidas em IV a fim de atestar a qualidade da ferramenta e identificar limitações.

5. ESTÁGIO DO TRABALHO

O trabalho encontra-se na etapa III, mais precisamente no projeto da solução. Até o momento, realizou-se um mapeamento sistemático (etapa I), identificando os principais esforços relacionados com a pesquisa (cerca de 231 artigos, extraído das principais conferências internacionais). Definiu-se o problema, os desafios e as metas da pesquisa, que estão apresentadas neste artigo (Seções 2 e 3). Além disto, analisou-se uma série de trabalhos relacionados (Seção 8) e definiu-se os requisitos-macro da solução (Seção 6).

6. SOLUÇÃO PROPOSTA

A solução proposta tem por objetivo analisar as aplicações do contexto Ginga-NCL, sob duas perspectivas: estrutural e comportamental, a fim de identificar os problemas existentes durante o processo de desenvolvimento. As subseções seguintes buscam apresentar detalhes da sua concepção.

6.1 Requisitos Funcionais

A solução foi idealizada para atender dois requisitos macros, permitir a análise estrutural (1) e comportamental (2) de uma aplicação desenvolvida para o contexto do Ginga-NCL. Desta forma, cada requisito macro será detalhado e validado, durante o processo de desenvolvimento, a fim de delimitar todas as funcionalidades da solução detalhadamente. As próximas seções descrevem brevemente cada requisito.

6.1.1 Análise Estrutural

Problemas estruturais são mais fáceis de serem identificados, geralmente as ferramentas de autoria fornecem esta funcionalidade de alguma forma, limitando-se evidenciar erros, léxicos, sintáticos

ou referenciais [12]. Especificamente, pretende-se que a análise estrutural identifique problemas de origem: **Léxica** – Referente à estrutura léxica definida pelo XML Schema; **Sintática** – Referente a hierarquia, obrigatoriedade e cardinalidade dos elementos e atributos; **Referencial** – Referente às relações de referência entre componentes do NCL; e **Contextual** – Referente às relações de contexto entre os elementos;

6.1.2 Análise Comportamental

Enquanto que, problemas comportamentais normalmente são identificados com o uso da simulação manual, procedimento ineficiente para este fim, pois pode necessitar de várias execuções para checar todos os comportamentos indesejáveis possíveis [12]. A solução proposta busca fornecer um modelo comportamental que apresente todos os estados possíveis da aplicação, que permita ao usuário, em cada estado, identificar: as interações possíveis, que resultem em uma mudança de estado; a descrição visual da tela; e os padrões de inconsistência conhecidos (heurísticas). Além disto, através de recursos interativos, pretende fornecer interfaces que possibilitem o usuário: Simular a execução da aplicação manualmente, navegando sobre o modelo comportamental; Executar fluxos de navegações relevantes, baseados nos caminhos possíveis originados pelo modelo comportamental; e Depurar o modelo comportamental, através do uso de *breakpoints* associados aos estados da aplicação.

6.2 Requisitos Não-Funcionais

Neste contexto, os requisitos não-funcionais são mais críticos e adicionam mais complexidade do que os funcionais, pois, além de poder inviabilizar o uso da ferramenta, eles afetam diretamente a sensação de qualidade do desenvolvedor ao utilizá-la. Dentre os requisitos-não funcionais pode-se destacar: ser completa; ser confiável; ser ágil; e ser usável. As próximas seções descrevem detalhadamente cada requisito, explicando a motivação, os desafios e como serão atendidos.

6.2.1 Ser Completa

A completude buscada pela solução, diz respeito à necessidade de suportar todas as regras normativas existentes. Pois é imprescindível identificar todos os problemas existentes em uma análise estrutural, além de ser essencial para a realização de uma análise comportamental, a fim de evitar a identificação de falsos comportamentos motivada pelo desconhecimento de alguma regra. As regras normativas não são bem estruturadas nas especificações, podendo estar descritas em várias sessões ou até mesmo em vários documentos, o que inviabiliza a checagem da completude por parte da ferramenta.

Dado que as assertivas representam unidades de funcionalidade, a ferramenta busca utilizar a abordagem orientada a assertivas no atendimento do requisito de completude, onde cada regra é associada unicamente a uma assertiva e tem como responsabilidade identificar a conformidade da aplicação (estrutural) e inferir comportamentos (comportamental) de acordo com a sua assertiva. Esta proposta permite avaliar a representatividade da ferramenta, pois relaciona unicamente as regras com as assertivas, além de possibilitar a utilização das 253 assertivas identificadas pelo ITU-T [5] como ponto de partida.

6.2.2 Ser Confiável

Para que a utilização da ferramenta seja produtiva, suas análises precisam ser confiáveis, evitando que o desenvolvedor precise recorrer à norma [1] para confirmar a existência de algum problema ou de algum comportamento evidenciado. Para isto,

além de ser necessário definir um processo de testes rígidos, é importante facilitar o acesso do usuário aos trechos específicos da norma relacionados com a análise, dado que oportunamente os usuários podem precisar de mais detalhes da especificação para entender alguns resultados.

A abordagem orientada a assertiva também agrega valor à confiabilidade da ferramenta, pois assim como [4], [5] e [8] utilizam esta abordagem para testar a conformidade dos interpretadores, pode-se utilizar as assertivas para testar a conformidade das análises. Além disto, pode-se apresentar a assertiva juntamente com os resultados das análises estruturais e comportamentais, permitindo que o usuário possa rastrear os trechos da norma que fundamentaram a análise da ferramenta.

6.2.3 Ser Ágil

A agilidade no contexto da solução é impactada por dois fatores: a velocidade na realização das análises e o tempo de resposta no seu acesso durante o processo de desenvolvimento. Em um cenário de desenvolvimento ágil, a adoção de novas ferramentas é bastante crítica, pois o custo da sua utilização pode inviabilizar seu uso. Com isso, tem-se como desafio propor uma solução que agregue valor à qualidade do produto, mas que não impacte negativamente na produtividade, quando inserido no processo de desenvolvimento.

O excesso de ferramentas independentes, com pouca integração entre elas, gera problemas de produtividade, necessitando de conhecimentos específicos para manusear cada uma, além de, eventualmente, necessitar de tratamentos manuais para integrá-las. Com isto, propõe-se a integração da solução com a plataforma Eclipse, normalmente utilizada como ambiente de desenvolvimento para o Gínga-NCL, quando integrada com os plug-ins apropriados (ex. NCL Eclipse [2] e Lua Eclipse⁵). Além disto, a integração com esta plataforma permite a utilização de recursos que possibilitam consumir as funcionalidades da ferramenta de forma automática, definindo visões dinâmicas para acesso das análises estruturais e comportamentais. Para viabilizar este recurso, propõe-se utilizar a estratégia de avaliação incremental orientada a mudanças [9], pois o processo de análise integral pode ser demasiadamente custoso, impossibilitando o acesso instantâneo das informações.

6.2.4 Ser Usável

É essencial para a solução atingir um grau de usabilidade aceitável, pois ela pode gerar uma elevada quantidade de informações, oriundas das avaliações estruturais e comportamentais, inviabilizando seu consumo. O perfil do usuário (desenvolvedor, testador etc.) é muito importante para este contexto, pois ele pode determinar a relevância e o nível de detalhamento das informações, contribuindo para a diminuição da curva de aprendizado, além de evidenciar a agregação de valor com o menor esforço de interação.

Para a ferramenta pretende-se utilizar a orientação ao perfil de usuário para criar interfaces mais atrativas e usáveis, impactando diretamente a forma de estruturar as informações. Este requisito é mais crítico na apresentação das avaliações comportamentais, pois uma aplicação pode gerar uma infinidade de comportamentos e estes devem ser omitidos ou detalhados de acordo com o propósito da análise. Propõe-se a utilização de autômatos finitos para definir os comportamentos possíveis, pois além de serem

indicados para este fim [7], possibilitam a utilização de técnicas de redução para ajustar o detalhamento das transições comportamentais [7].

7. AVALIAÇÕES PROPOSTAS

Considerando que a solução proposta se insere no contexto de desenvolvimento de aplicações, auxiliando na identificação e resolução de problemas estruturais e comportamentais, inicialmente, devido à criticidade dos requisitos não-funcionais, é importante atestar o atendimento de cada um através de experimentos formais. E, de forma complementar, é essencial avaliar a eficiência da solução em relação à análise manual e aos trabalhos relacionados mais relevantes. Para isto é importante:

- Avaliar a confiabilidade e completude da solução, através da análise de uma base de aplicativos composta por: aplicações reais com problemas existentes conhecidos, aplicações manipuladas contendo os mais diversos tipos de erros (estruturais e comportamentais) e aplicações corretas (sem erros). Afim de confrontar seus resultados com os da execução em set-top-boxes reais e emuladores relevantes; e
- Avaliar o desempenho e a usabilidade da solução, durante o processo de criação e manutenção de aplicativos. Afim de confrontá-lo com o desempenho de outros trabalhos relacionados relevantes e com o desempenho da análise manual.

8. TRABALHOS RELACIONADOS

O trabalho apresentado tem por objetivo identificar os problemas sintáticos, estruturais e comportamentais presentes nas aplicações desenvolvidas para o Gínga-NCL. Nesse contexto, podemos destacar os seguintes trabalhos: dos Santos *et al.* [10], Honorato e Barbosa [3], Picinin Jr. *et al.* [7], e Santos *et al.* [9].

Em Santos *et al.* [9], é apresentado um processo de validação incremental do NCL, que realiza validação sintática e estrutural apenas nas partes do código que foram modificadas ou impactadas desde a última validação. Tal abordagem é avaliada através de um teste de desempenho que a compara com a validação completa. O foco principal é diminuir o esforço de validação, identificando o menor conjunto de segmentos a ser validado, composto pelos trechos que sofreram modificações diretas, ou pelos que possivelmente foram afetados. Além disto, permite a configuração automática do tipo de validação a ser feita, sintática ou estrutural, por exemplo, baseado no conjunto de elementos a serem validados.

Honorato e Barbosa [3] apresentam o NCL-Inspector, que apoia a autoria de aplicações NCL, através da inspeção do código, identificando erros e sugerindo melhorias. Para isto utiliza uma base de regras, que pode ser expandida, onde cada regra é composta por uma condição (padrão de estrutura) e uma ação. A inspeção é realizada em uma árvore sintática abstrata (*Abstract Syntax Tree - AST*) que representa o NCL, buscando violações das regras existentes.

Dos Santos *et al.* [10] e Picinin Jr. *et al.* [7] se assemelham no ponto em que utilizam uma abordagem orientada a modelos para analisar documentos NCL. Além disto, representam a aplicação através de um modelo simplificado que utiliza fórmulas em lógica temporal para checar a existência de comportamentos indesejáveis. Porém, [10] representa o NCL através de um modelo formal e o utiliza para verificar algumas propriedades estruturais,

⁵ luaeclipse.luaforge.net

enquanto que [7] realiza apenas análises comportamentais representando o documento NCL através de uma linguagem intermediária para posteriormente transformar a representação em uma rede de Petri, que destaca os comportamentos possíveis.

A Tabela 1 evidencia as principais características de cada trabalho relacionado. É importante destacar que todos os trabalhos buscam estruturar suas regras normativas e possibilitam sua criação/atualização pelo usuário. (*) Dentre os trabalhos que realizam análise comportamental, apenas um fornece uma análise comportamental real, os outros se limitam a identificar heurísticas existentes no fluxo da aplicação. Além disto, vale ressaltar que nenhum trabalho avaliado fornece detalhes de conformidade ou completude, negligenciando a referência das análises com a especificação da linguagem.

Tabela 1. Quadro comparativo das características dos trabalhos relacionados

| Característica | [9] | [3] | [10] | [7] |
|--|-----|-----|------|-----|
| Realiza análise estrutural | ✓ | ✓ | ✓ | ✗ |
| Realiza análise comportamental | ✗ | ✓* | ✓* | ✓ |
| Fornecer detalhes da conformidade com a norma | ✗ | ✗ | ✗ | ✗ |
| Fornecer detalhes da completude da análise | ✗ | ✗ | ✗ | ✗ |
| Utiliza um modelo simplificado para representar a aplicação | ✗ | ✓ | ✓ | ✓ |
| Utiliza um modelo estruturado para representação das regras normativas | ✓ | ✓ | ✓ | ✓ |
| Permite a criação/atualização das regras | ✓ | ✓ | ✓ | ✓ |
| Possibilita a integração com um ambiente de desenvolvimento | ✓ | ✗ | ✗ | ✓ |
| Define estratégias para reutilizar análises | ✓ | ✗ | ✗ | ✗ |

8.1 Rastreamento de regras normativas

Em um processo de identificação de problemas, que busca evidenciar infrações e interpretações incorretas da especificação, faz-se necessário apresentar seus resultados de forma precisa e confiável, referenciando cada trecho da especificação que os justificam. Nesse contexto, podemos destacar alguns trabalhos que buscam estruturar especificações descritivas para objetivos diversos, tais como [4], [5] e [8].

A ITU-T [5] define uma especificação padrão, orientada a assertivas, para realização de testes de conformidade no contexto do Ginga-NCL. Esta especificação é organizada em três categorias: assertivas, instruções de testes e casos de testes. As assertivas são compostas por um identificador, por uma descrição, pelas referências e pelo alvo (componente NCL). Pinheiro *et al* [8] especificam um conjunto contendo 253 assertivas referentes a ITU-T [5], utilizando as assertivas da especificação padrão para criação de seus casos de testes. Ainda com o objetivo de especificar testes, [4] define um processo para escrita de suítes de testes para o HTML, para isto também utiliza o paradigma orientado a assertivas. Neste paradigma, a primeira etapa é caracterizada pela identificação de assertivas testáveis, agrupadas por categorias que auxiliem a sua organização, tais como: seção, subseção e funcionalidades.

A orientação a assertivas é indicada para a estruturação de especificações descritivas, pois representam unidades de funcionalidade ou comportamento extraídas das proposições normativas, que constituem uma especificação modular, estruturada e completa. A fim de manter a rastreabilidade, as

assertivas mantêm a associação com os trechos da norma que fundamentaram sua criação.

9. REFERÊNCIAS

- [1] ABNT NBR 15606-2:2011. Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiofusão digital Parte 2: Ginga-NCL para receptores fixos e móveis - Linguagem de aplicação XML para codificação de aplicações, 2011
- [2] Azevedo, R. G. de A.; Teixeira, M. M.; Soares Neto, C. S.. NCL Eclipse: Ambiente Integrado para o Desenvolvimento de Aplicações para TV Digital Interativa em Nested Context Language. SBRC, Recife, 2009.
- [3] Honorato, Gustavo de Sá Carvalho; e Barbosa, Simone Diniz Junqueira. NCL-inspector: towards improving NCL code. In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10). 2010
- [4] HTML Test Suite Documentation
- [5] ITU-T - Conformance testing specification for H.761
- [6] Petersen, K.; Feldt, R., Mujtaba, S.; Mattsson M.. Systematic Mapping Studies in Software Engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering, Australia, 2008.
- [7] Picinin Jr, Delcino; Farines, Jean-Marie; e Koliver, Cristian. An approach to verify live NCL applications. In Proceedings of the 18th Brazilian symposium on Multimedia and the web (WebMedia '12). 2012.
- [8] Pinheiro, Cláudio F.; Filho, Eddie B. L.; Oliveira, Rodrigo R.; Cavalcante, Anastácio A. M.; Klehm, Volnei S.; Pereira, Daniel P.; e Melo, Hallyson L. M.. Uma Proposta de Suíte de Testes de Conformidade para o Ginga-NCL. In XXX Simpósio Brasileiro de Telecomunicações (SBRT'12). 2012.
- [9] Santos, Rodrigo C.M.; Cerqueira Neto, Jose R; Soares Neto, Carlos S.; e Teixeira, Mario M.. Incremental validation of digital TV applications in nested context language. In Proceedings of the 10th European conference on Interactive tv and video (EuroITV '12). 2012.
- [10] Santos, J.A.F. dos, Braga, Christiano, e Muchalut-Saade, Débora C.. Automating the analysis of NCL documents with a model-driven approach. In Proceedings of the 19th Brazilian symposium on Multimedia and the web (WebMedia '13). 2013.
- [11] Santos, C.A.S.; Ferraz, C.A.G.; Lenz, A.R.; J.S. Silva. Desenvolvimento de Aplicações Sensíveis ao Contexto da Programação diante das Não Conformidades dos Sinais das TV Brasileiras. Revista de Radiodifusão, v. 7, p. 50-59, 2013.
- [12] Santos, C. A. S.; Lenz, A. R. Lenz. Non-conformance and Problems in Digital TV Signals Transmitted by Brazilian Broadcasters and Their Impact on the Users and TV Application. IEEE Latin America Transactions. 2014.
- [13] Soares, L. F. G.; Moreno, M. F.; Neto, C. de S. S.; Moreno, M. F.. Ginga-NCL : Declarative Middleware for Multimedia IPTV Services. IEEE Communications Magazine, 2010.
- [14] Soares, L. F. G.; Moreno, M. F.; Costa, R. M. R.; Moreno, M. F.. Towards the convergence of digital TV systems. Journal of Internet Services and Applications. 2010.