Science of Computer Programming 107–108 (2015) 64–92

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

A rewriting logic semantics for NCL[☆]

Joel dos Santos[a,b,*], Christiano Braga[b], Débora C. Muchaluat-Saade[a,b]

[a] MídiaCom Lab, Universidade Federal Fluminense, Brazil
[b] Institute of Computing, Universidade Federal Fluminense, Brazil

ARTICLE INFO

Article history:
Received 16 May 2014
Received in revised form 24 April 2015
Accepted 28 April 2015
Available online 6 May 2015

Keywords:
Multimedia document validation
Multimedia authoring
Rewriting theory
Maude
NCL

ABSTRACT

The Nested Context Language (NCL) is a multimedia authoring language that is part of the standard for digital television in Brazil and in Latin America and the ITU standard for IPTV services. To properly support authors in the development of NCL applications, it is important to provide tools with validation capabilities. The validation of multimedia documents indicates to the author possible inconsistent points in a given document. Thus the author has the possibility of correcting those inconsistencies before making the multimedia document available for the final user. The validation approach presented in this work considers a set of desirable document properties based on related work published in the literature. Since NCL applications may be understood as finite transition systems, standard model-based validation techniques directly apply. This paper discusses a formalization of NCL semantics with rewriting theory. Two theories are given to represent (i) NCL's static semantics and (ii) NCL's dynamic semantics. In the former, NCL structural properties are validated through equational reduction, while in the latter, NCL expected behaviors are validated through searching.

1. Introduction

The declarative programming paradigm focuses on the description of the author's intention, instead of decomposing programs in a sequence of steps. Such an approach is supported in the multimedia scenario, to ease the task of authoring multimedia documents. Such a goal is important since multimedia documents can be used in different areas, such as web, digital TV and IPTV; and by different author profiles, such as developers and content producers.

A multimedia document is possibly composed by several media objects, such as audio, video, image or even other multimedia documents. Media objects are organized in a multimedia document by declarations provided by the author. Such declarations specify temporal synchronization relationships among media objects, by defining the temporal order of media objects, such as user interaction with (part of) media objects, which are called *anchors* in NCL.

Nested Context Language (NCL) [1,2] is a declarative multimedia document authoring language, which allows for the specification of multimedia documents. NCL is part of the standard for digital television (DTV) in Brazil and Latin America [1]. It is also part of the ITU standard for IPTV services [2]. NCL is an XML-based language, i.e. it uses the Extensible Markup Language (XML) [3] as concrete syntax.

[☆] This work was partially funded by FAPERJ and CNPq.

[*] Corresponding author at: Institute of Computing, Universidade Federal Fluminense, Brazil.
E-mail addresses: joel@midiacom.uff.br (J. dos Santos), cbraga@ic.uff.br (C. Braga), debora@midiacom.uff.br (D.C. Muchaluat-Saade).

http://dx.doi.org/10.1016/j.scico.2015.04.006
0167-6423/© 2015 Elsevier B.V. All rights reserved.

The creation of an NCL document may become more complex when the document grows in the number of media objects and/or synchronization relationships. Media objects may be organized in a complex structure or temporal order, which demands more synchronization relationships and, therefore, more declarations from the author. Taking into account the XML structure needed to represent such documents, more complex documents imply more XML elements.

The author of such complex documents may produce errors during authoring. We divide those errors in two kinds, called *structural problems* and *undesired behaviors*. Structural problems arise when a multimedia document does not follow the grammar of the given multimedia authoring language and does not have certain properties regarding its element nesting structure. Undesired behaviors arise when the computation induced by the multimedia document, when executed, produces a possibly unexpected presentation. Examples of undesired behaviors are when parts of the document are not executed, during document presentation, or do not end their presentation. We call it possibly unexpected since such behavior may be expected by the author, however different approaches in the literature also identify those behaviors as undesired [4–6].

Authoring tools with validation capability are necessary to avoid authoring errors or at least provide the author a way to find them prior to deployment. We call multimedia document *validation* the process of investigating desired properties in a given document in order to guarantee an error-free presentation, i.e., its presentation without errors or unexpected behaviors. In [7] we have identified a set of properties for NCL documents that appear to be desirable by the multimedia document authoring community. They are classified as structural and behavioral properties. Structural properties, part of NCL's static semantics, specify the authoring language syntax rules by means of invariants. Behavioral properties, part of NCL's dynamic semantics, are those used for representing desired behaviors in a document's presentation.

As part of our project [8] that aims at developing formal support for NCL document reasoning, we define a Rewriting Logic theory [9] for the specification of NCL's semantics. We propose Rewriting Logic as a semantic framework for NCL because the semantics of NCL documents is directly captured by a Rewriting Logic theory. Essentially, given an NCL document $N = (A, L)$ where A is a set of anchors and L a set of links, a Rewriting Logic theory $\mathcal{N} = (\Sigma, E, R)$ represents N by declaring the anchors of A in Σ, links L as equations E and the different possible interactions a user may perform as rules in R. The executability of Rewriting Logic theories in the Maude system [10] allows for the automatic verification of many desired properties of N through search or model checking. Equations represent structural properties and temporal logic formulas represent behavioral properties, which are verified in the Maude system [10] by equational reduction and model checking, respectively. In order for us to prove properties about our mapping from NCL to Rewriting Logic, we first define (in Section 3) a set-theoretic semantics for NCL as, to the best of our knowledge, no formal meaning to NCL documents is widely accepted. Then, after discussing NCL's rewriting semantics (in Section 4), we prove that both the contents of a document and its presentation are preserved by our mapping with respect to our set-theoretic semantics. Finally, with a sound representation of NCL documents (considering the set-theoretic semantics), we may confidently prove properties about NCL documents using our rewriting semantics.

This paper is a significantly revised and extended version of [11], presented in XVI Brazilian Symposium on Formal Methods in Brasília, DF, Brazil, and published in its proceedings. Sections 3, 4 and 5 in [11] were completely rewritten. They now discuss in detail the syntax and semantics of NCL in a set-theoretic setting, recall Rewriting Logic elements in sufficient depth to discuss our proposed semantics, and present a *new* Rewriting Logic semantics for NCL. Our new semantics relies on the so called set-rewriting theories that allows for rewriting *modulo* axioms of associativity, commutativity and identity. Essentially, using set-rewriting theories, a non-deterministic NCL core Rewriting Logic specification is used to rewrite a term, representing a document state, modulo equations that specify links of the given NCL document. Our new semantics not only gives a *clearer* semantics for NCL documents but also provides a more *efficient* one, with respect to our original proposal.

The remainder of this paper is structured as follows. Section 2 presents related work considering the validation of multimedia documents. Section 3 introduces the NCL language, its syntax and semantics. Section 4 describes the rewriting logic theories we define for representing the semantics of NCL. Section 5 presents how the rewriting logic theories representing NCL's semantics may be used for the validation of NCL documents. Section 6 concludes this paper highlighting its contributions and pointing to future work.

2. Related work

The literature is rich on the discussion about the validation of multimedia documents. In general, it describes *either* the behavioral or structural validation. Therefore, this section presents related work separating contributions on the structural validation from those focusing on the behavioral validation.

2.1. Structural validation

Honorato and Barbosa in [12] present the NCL-Inspector tool. This tool, based on other tools for code quality critique, supports the author in terms of code quality. With this tool, moreover the possibility of verifying the NCL code searching for coding problems, it is possible to suggest modifications regarding best programming practices. The code validation, or inspection, is done following a set of rules, written in Java or XSLT, forming a rule repository. Each rule presents an NCL code pattern and an action performed when that pattern is found. For the inspection of an NCL document, NCL-Inspector parses the document. After that step, the tool creates an abstract syntax tree that represents the NCL document being inspected. Then it walks through the tree searching for violations of the existent rules. The violations found are presented to the user