A Model-driven Approach for the Analysis of Multimedia Documents

Joel dos Santos[1], Christiano Braga[2], and Débora Muchaluat-Saade[1]

[1] Laboratório MídiaCom
{joel,debora}@midiacom.uff.br
[2] Language-oriented software engineering research group
cbraga@ic.uff.br
Computer Science Department
Universidade Federal Fluminense

Abstract. This paper proposes a model-driven approach for the analysis of multimedia documents. Structural and behavioral properties of a multimedia document are verified thus guaranteeing its well-formedness and conformance before deployment. Multimedia documents are interpreted as object model instances of a multimedia document metamodel. Structural properties are verified using consistency reasoning over an ontology representation of the given object model together with OCL invariant validation (i.e., the application of OCL invariants to the given object model). Behavioral properties are verified through model checking on the transition system associated to the given multimedia document. Both metamodel and user-defined behavioral properties are verified.

1 Introduction

Multimedia documents describe applications as a set of components, which represent media objects, and relationships among them, which define temporal and real-time constraints over components. Declarative *authoring* languages may simplify the definition of multimedia documents since they emphasize the description of a document rather than the implementation of its presentation.

Large multimedia documents, with many components, organized within quite rich and complex structures, may be ill-formed and fall victim of conflicting relationships, leading to an application whose presentation is not what the author desires. An example of an ill-formed document includes cycles in composition nesting, where a composition is a group of components and relationships. Examples of undesired behavior (e.g. [7, 12, 13]) are the non-termination and/or unreachability of parts of a given document and the concurrent use of system resources, like an audio channel or a space at the screen. Usually, authors test their documents by executing them in an attempt to identify undesired behaviors, an approach which is not *complete* since not all possible behaviors may be explored on potentially *ill-formed* models.

Model-driven development (MDD) is a software development approach where models are the main artifacts of the development process. Models are represented as instances of metamodels, which describe the syntax of a modeling language, and may be used to derive different software artifacts, such as code in a programming language or even other models in different abstraction levels. This derivation process is called model transformation, which relates modeling languages through metamodels. Under that perspective, MDD is seen as a transformation between modeling languages applied to particular models, as shown in Figure 1,

$$m \in M \xrightarrow{parsing} \hat{m} \in \mathcal{M} \xrightarrow{\tau} \hat{n} \in \mathcal{N} \xrightarrow{pretty\text{-}printing} n \in N$$

Fig. 1. The application of a model transformation $\tau : \mathcal{M} \to \mathcal{N}$

where M represents the concrete and $\mathcal{M}$ the abstract syntax of a source modeling language, N represents the concrete and $\mathcal{N}$ the abstract syntax of a target modeling language and τ the transformation between $\mathcal{M}$ and $\mathcal{N}$. The operation *parsing* represents a mapping where a model m produces an instance $\hat{m}$ of $\mathcal{M}$ and *pretty-printing* represents the inverse mapping. The notation $m \in M$ denotes that model m is (syntactically) *well-formed* with respect to metamodel M.

The objective of MDD is to increase the abstraction level of the development process, so that authors may focus on modeling rather than implementation. In the context of multimedia documents, this is precisely the objective of NCL [9] and other multimedia authoring languages such as SMIL [15] and HTML5 [16]. Being *language-driven*, as shown in Figure 1, MDD fits very nicely with the development of multimedia applications in general and in particular with the validation of multimedia documents.

As previously mentioned, the verification of behavioral properties, like the ones presented before, is an important task in the development of multimedia applications, since it guarantees important properties of multimedia documents. Automaton based techniques fit well in this type of verification through model-checking [5] which essentially defines a decision procedure for temporal formulae. The application of model checking to multimedia documents thus requires a formalization of its behavior as a labeled transition system (LTS) and a proper encoding of the desired behavioral properties as temporal formulae.

A question that arises then is: *how to guarantee that the model being verified correctly represents the multimedia document?*

Figure 2 shows MDD with the so-called transformation contracts [2–4]. A transformation contract is a specification of a model transformation defined as a model that relates the metamodels of two modeling languages. Formally, a transformation contract $\mathcal{K}$ for a model transformation $\tau : \mathcal{M} \to \mathcal{N}$ is the disjoint union $\mathcal{M} \bowtie_{\mathcal{A}} \mathcal{N}$ of the metamodels of the modeling languages together with associations in $\mathcal{A}$ between the model elements of $\mathcal{M}$ and $\mathcal{N}$. The associations in $\mathcal{A}$ may be constrained by different kinds of properties, either structural or behavioral. The idea is that, following a *design by contract* style, every time a model transformation $\tau : \mathcal{M} \to \mathcal{N}$ is applied to a model $\hat{m} \in \mathcal{M}$, first the properties $P_{\mathcal{M}}$ of the source metamodel are verified in $\hat{m}$, then properties $P_{\mathcal{N}}$ of the target metamodel are verified in the generated model $\hat{n} = \tau(\hat{m})$, together with the properties $P_{\mathcal{K}}$ attached to the relations specified by $\mathcal{A}$, which must hold on the joined model $k = \hat{m} \bowtie_l \hat{n}$ with $l \in \mathcal{A}$. The notation $m \models P_{\mathcal{M}}$ means that the properties $P_{\mathcal{M}}$ hold in model m, that is, m *is in conformance* with $\mathcal{M}$.

$$m \in M \xrightarrow{parsing} \begin{array}{c} \hat{m} \in \mathcal{M}, \\ \hat{m} \models P_{\mathcal{M}} \end{array} \xrightarrow{\tau} \begin{array}{c} \hat{n} \in \mathcal{N}, \\ \hat{n} \models P_{\mathcal{N}}, \\ k \models P_{\mathcal{K}} \end{array} \xrightarrow{pretty\text{-}printing} n \in N$$

Fig. 2. MDD with transformation contracts

An MDD process for the analysis of multimedia documents would refine the process of Figure 2 by understanding $\mathcal{M}$ as the metamodel of a multimedia authoring language, such as NCL, and $\mathcal{N}$ as the metamodel of the specification language of a formal verification framework such as the specification language of a model checker.

This work proposes a transformation contract approach for the analysis of multimedia documents. Different verification techniques shall be used to analyze multimedia documents: (i) Consistency reasoning with description logic [1] will be used for verifying document consistency together with Object Constraint Language (OCL) invariant execution; and (ii) Linear Temporal Logic model checking appears to be the appropriate reasoning technique for behavioral properties of multimedia documents.

This work contributes with a general framework, with tool support, capable of analyzing different types of multimedia documents using different analysis (that is, verification and validation) techniques. Our proposal uses a language-driven approach where the authoring language semantics is represented by a general model (called SHM - Simple Hypermedia Model) where structural and behavioral properties are verified. In this paper we outline our approach and discuss preliminary results achieved with a prototype of the tool.

The remainder of this paper is organized as follows. Section 2 presents the state-of-the-art on multimedia document analysis. Section 3 discusses the proposed solution for multimedia document analysis. Section 4 discusses the current state of the multimedia document analysis project illustrating preliminary results. Section 5 finishes this paper presenting the next steps of this work.

2 State-of-the-art on multimedia document analysis

Santos et al. [13] presented an approach for the analysis of multimedia documents by translating it into a formal specification, in that case, into RT-LOTOS processes, using general mapping rules. The modularity and hierarchy of RT-LOTOS allows the combination of processes specifying the document presentation with other processes modeling the available platform.

The verification consists in the interpretation of the minimum reachability graph built from the formal specification, in order to prove if the action corresponding to the presentation end can be reached from the initial state. Each node in the graph represents a reachable state and each edge, the occurrence of an action or temporal progression. When a possible undesired behavior is found, the tool returns an error message to the author, so he can repair it. The tool in [13] could analyze NCM [14] and SMIL [15] documents.

Na and Furuta, in [12], presented caT (context aware Trellis), an authoring tool based on Petri nets. caT supports the analysis of multimedia documents by building the reachability tree of the analyzed document. The author defines limit values for the occurrence of dead links (transitions that may not be triggered), places with token excess, besides other options, as the analysis maximum time. The tool investigates the existence of a terminal state, i.e., whether there is a state where no transitions are triggered. It also investigates the limitation property, i.e., if no place in the net has an unlimited number of tokens and the safeness property, i.e., if each place in the net has a token. The limitation analysis is important since tokens may represent scarce system resources.

Oliveira et al., in [7], presented HMBS (Hypermedia Model Based on Statecharts). An HMBS multimedia application is described by a statechart that represents its structural hierarchy, regarding nodes and links, and its *human-consumable* components. Those components are expressed as information units, called pages and anchors. The statechart execution semantics provide the application navigation model. A statechart state is mapped into pages and transactions and events represent a set of possible link activations.

The statechart reachability tree for a specific configuration may be used to verify if any page is unreachable, by verifying the occurrence of a state s in one of the generated configurations, which indicate that the page is visible when the application navigation starts in the initial state considered. In a similar manner, it is possible to determine if a certain group of pages may be seen simultaneously searching state configurations containing the states associated to those pages. The reachability tree also allows the detection of configurations from which no other page may be reached or that present cyclical paths.

Júnior et al., in [10], also present the verification of NCL documents through a model-driven approach. The verification is also achieved by transforming an NCL document into a Petri Net. This transformation is done in two steps. The first step transforms the NCL document into a language called FIACRE, representing the document as a set of components and processes. Components represent media objects and compositions and processes represent the behavior associated to components. The second step transforms the FIACRE representation into a Petri Net. The verification uses a model-checking tool and temporal logic formulae to represent the behavior the author wants to verify. Once this work is very recent, the automation of that approach is a future work.

Our work contributes to the state-of-the-art with a *general* approach that can be used with different multimedia authoring languages.

3 A model-driven approach to multimedia document analysis

We propose the use of the transformation contracts approach to analyze multimedia documents. Figure 3 refines Figure 2 and illustrates our approach pictorially with NCL as the multimedia authoring language and Maude [6] as the specification language for formalizing multimedia documents. Informally, Maude modules are produced from NCL documents and the behavioral properties are represented as LTL formulae which are verified using the Maude model checker.

An important element of our approach is the so-called modeling language for the Simple Hypermedia Model (SHM) [8]. SHM models are important for two reasons: (i) they give formal meaning to NCM models, and (ii) should be a *general* formal representation for multimedia documents. SHM models are essentially transition systems that have basic elements to represent multimedia documents such as *anchors* as states, *events* as actions and *links* as transitions.From SHM models we could produce representations in different formalisms such as Maude or SMV [11]. Behavioral properties of well-formed models that hold the structural properties of a given authoring language are then checked at the concrete level such as Maude or SMV.

Let us go through each step of Figure 3. First, an NCL document is *parsed* into an NCM [14] model. (NCM is the conceptual model that NCL documents are based on and may be understood as its abstract syntax.) Thus, given an NCL document d, if $(\hat{d} = parse(d)) \models P_{\mathcal{NCM}}$, that is, if the structural properties of NCM hold in $\hat{d}$ (such as non-circular nested compositions) then a model transformation $\tau_{\mathcal{NCM}}$ is applied on $\hat{d}$. Given that a proper SHM model $\hat{s}$ is produced by the application of the transformation contract from NCM to SHM, that is, essentially, its states are built properly from anchors, actions properly built from events and transitions properly built from links, a concrete representation of $\hat{s}$ may be produced in the specification language of the model checker, such as Maude.

$$d \in NCL \xrightarrow{parsing} \begin{array}{l}\hat{d} \in \mathcal{NCM},\\ \hat{d} \models P_{\mathcal{NCM}}\end{array} \xrightarrow{\tau_{\mathcal{NCM}}} \begin{array}{l}\hat{s} \in \mathcal{SHM},\\ \hat{s} \models P_{\mathcal{SHM}},\\ k \models P_{\mathcal{K}}\end{array} \xrightarrow{pretty\text{-}printing} m_d \in Maude$$

Fig. 3. A transformation contract approach to Maude theories from NCL documents

Given m_d, which is well-formed and in conformance with $K = \mathcal{NCM} \bowtie_{\mathcal{A}} \mathcal{SHM}$, one can now verify with a model-checker the temporal formulae that represent the behavioral properties exemplified at the beginning of Section 1 (such as unreachability of document parts) and document specific properties, defined by the document author and transformed into temporal formulae. Counterexamples produced by the model-checker, which are essentially traces that do not have the desired temporal formulae, may be presented back to the document author as sequences of links representing SHM transitions that correspond to transitions (or rewrites, in the case of Maude) of the faulty path encountered by the model checker. This process is illustrated pictorially in Figure 4, where

$$NCL\ author \underset{l \in (NCL_{Links})^*}{\overset{d \in NCL}{\rightleftarrows}} NCL\ Analyzer = \tau(parse(d)) \vdash modelCheck(s_0, \phi)$$

Fig. 4. NCL Analyzer

NCL Analyzer is the tool that essentially invokes the Maude model checker, represented in Figure 4 by the command *modelCheck*, which checks for the property ϕ (a conjunction of the behavioral properties together with author-defined properties) using the specification (actually, rewrite theory) given by $\tau(parse(d))$ using s_0 as initial state (specified by the initial conditions of document d).

As mentioned before, SHM is intended to be a general multimedia model. The verification of multimedia documents specified with languages different from NCL, such as SMIL and HTML5, would require transformations from the abstract syntax of those languages to *SHM* together with a proper mapping from counter-examples of the chosen model-checker to the authoring language. The remaining of the analysis process is reused among those different languages.

We have a first attempt at SHM and a prototype tool that transforms NCL to Maude modules. Section 4 briefly discusses preliminary results.

4 Preliminary results

Part of the proposed solution is prototyped in a tool presented in [8], where the first author, under the supervision of the remaining authors, proposed an implementation of a transformer from NCL documents to Maude modules. With that prototype it was possible to analyze structural and behavioral properties of NCL documents. Besides, the prototype gives us the intuition that the proposed solution seems to be appropriate.

The prototype was used in several small experiments with simple documents. Besides it was used with two non-trivial documents created by the Brazilian Digital TV community. A description of the two documents ("First João" and "Live More") and their results are presented here.

"First João" is an interactive TV application that presents an animation inspired in a chronicle about a famous Brazilian soccer player named Garrincha. It plays an animation, an audio and a background image. At the moment Garrincha dribbles the opponent, a video of kids performing the same dribble is presented and when his opponent falls on the ground, a photo of a kid in the same position is presented. The user may interact with the application pressing the red key at the moment a soccer shoes icon appears. The animation is resized and a video of a kid thinking about shoes starts playing.

This document was deployed by the authors of the NCL language as a sample document. As expected, the document is consistent with respect to the structural properties ($P_{\mathcal{NCM}}$), defined taking into account the NCM grammar, and the behavioral properties ($P_{\mathcal{SHM}}$), from the set of parameterized properties. It was possible to verify that every anchor is reachable and has an end. Besides, the document as a whole ends.

"Live More" is an application that presents a TV show discussing health and welfare. Once the TV show starts playing, an interaction icon appears. If the user presses the red key of the remote control, four different food options appear. The user can choose a dish by pressing one of the colored keys of the remote control. When a dish is chosen, the TV user is informed about the quality of his choice, telling whether there are missing nutrients or nutrients in excess.

This document is consistent with respect to the structural properties ($P_{\mathcal{NCM}}$). However, the document is not consistent with respect to the behavioral properties ($P_{\mathcal{SHM}}$). It was possible to verify that once a dish is chosen, the anchor representing the chosen dish and its result do not end, and consequently the document as a whole.

The proposed prototype allows NCL document authors to verify if their document fails in one of the common undesired properties, besides validating the document structure. From the tests done with NCL documents it was possible to identify refinements in our Maude specification of $\mathcal{SHM}$. Such refinements and open issues are addressed in the next section.

5 Conclusion

In this paper we presented an approach for the analysis of multimedia documents and a prototype tool that partially implements it. This section discusses future directions to our research project.

We are currently working on a refinement of the specification for $\mathcal{SHM}$ in [8], its Maude representation (to improve the efficiency of model checking it) and on a formal proof for the transformation $\tau_{\mathcal{NCM}}$.

An important future work it to evaluate the generality of our approach, exploring mappings from different authoring languages to $\mathcal{SHM}$, as indicated in the end of Section 3.

Our preliminary results consider predefined properties representing patterns of behavior of multimedia documents (see Section 3.) We plan to incorporate user-defined behavioral properties, by allowing the author to define such properties in a *structured* natural language (English, for example) that could be translated to LTL formulae.

We also consider evaluating the usability of the tool resulting from this project using human-computer interaction techniques.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2003.

2. C. Braga. A transformation contract to generate aspects from access control policies. *Journal of Software and Systems Modeling*, 10(3):395–409, 2010.
3. C. Braga, R. Menezes, T. Comicio, C. Santos, and E. Landim. On the specification verification and implementation of model transformations with transformation contracts. In *14th Brazilian Symposium on Formal Methods*, volume 7021, pages 108–123, 2011.
4. C. Braga, R. Menezes, T. Comicio, C. Santos, and E. Landim. Transformation contracts in practice. *IET Software*, 6(1):16–32, 2012.
5. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* The MIT Press, 2000.
6. M. Clavel, S. Eker, F. Durán, P. Lincoln, N. Martí-Oliet, and J. Meseguer. *All about Maude - A High-performance Logical Framework: how to Specify, Program, and Verify Systems in Rewriting Logic.* Springer-Verlag, 2007.
7. M.C.F. de Oliveira, M.A.S. Turine, and P.C. Masiero. A statechart-based model for hypermedia applications. *ACM Transactions on Information Systems*, 19(1):52, 2001.
8. J. A. F. dos Santos. Multimedia and hypermedia document validation and verification using a model-driven approach. Master's thesis, Universidade Federal Fluminense, 2012.
9. ITU. Nested Context Language (NCL) and Ginga-NCL for IPTV services. http://www.itu.int/rec/T-REC-H.761-200904-S, 2009.
10. D. P. Júnior, J. Farines, and C. A. S. Santos. Uma abordagem MDE para Modelagem e Verificação de Documentos Multimídia Interativos. In *WebMedia*, 2011. in Portuguese.
11. K.L. McMillan. *Symbolic model checking: an approach to the state explosion problem.* Kluwer Academic Publishers, 1993.
12. J.C. Na and R. Furuta. Dynamic documents: authoring, browsing, and analysis using a high-level petri net-based hypermedia system. In *ACM Symposium on Document engineering*, pages 38–47. ACM, 2001.
13. C.A.S. Santos, L.F.G. Soares, G.L. de Souza, and J.P. Courtiat. Design methodology and formal validation of hypermedia documents. In *ACM International Conference on Multimedia*, pages 39–48. ACM, 1998.
14. L. F. G. Soares, R. F. Rodrigues, and D. C. Muchaluat-Saade. Modeling, authoring and formatting hypermedia documents in the HyperProp system. *Multimedia Systems*, 2000.
15. W3C. Synchronized Multimedia Integration Language - SMIL 3.0 Specification. http://www.w3c.org/TR/SMIL3, 2008.
16. W3C. HTML5: A vocabulary and associated APIs for HTML and XHTML. http://www.w3.org/TR/html5/, 2011.