Verifying Hypermedia Applications by Using a MDE Approach

Cristian Koliver - Federal University of Santa Catarina,
Delcino Picinin Júnior - Federal Institute of Santa Catarina,
Celso A. S. Santos - Federal University of Espirito Santo and
Jean-Marie Farines - Federal University of Santa Catarina

BRASIL

Valencia, 29 september 2014

Summary

1. Introduction
2. The Proposed Design Method
3. Formal Verification
4. Toolchain
5. Conclusions

Summary

1. Introduction
   - Hypermedia Document Design
   - Design Solutions
   - Main Challenges
2. The Proposed Design Method
   - Design Method
   - Transformation Step: from Designer Representation to Verification Representation
3. Formal Verification
   - Observers
   - Verification
   - Verification in Practice
4. Toolchain
   - Toolchain Associated to the Proposed Method
5. Conclusions
   - Performance Analyze
   - Conclusions

Hypermedia Document Design

Some Issues:

- Hypermedia Document Requirement:
    - time constraints
    - spatial constraints
    - user interactions
- Live Design
- Designer, publicist and journalist: limited knowledge in computing

Drawback

- Undesirable behaviors introduced during the creation of document

Summary

1. Introduction
   - Hypermedia Document Design
   - Design Solutions
   - Main Challenges
2. The Proposed Design Method
   - Design Method
   - Transformation Step: from Designer Representation to Verification Representation
3. Formal Verification
   - Observers
   - Verification
   - Verification in Practice
4. Toolchain
   - Toolchain Associated to the Proposed Method
5. Conclusions
   - Performance Analyze
   - Conclusions

Design Solutions

Usual Solution

- Test of all possible behaviors
    - A lot of work, costly
    - Non-exhaustive
    - Inappropriate in live editions (due to the time required)

Proposed Solution: Three-step method based on verification

1. Modeling/Edition:
    - Hypermedia languages (NCL and SMIL)
2. Transformation:
    - **From** Hypermedia Document language **to** Formal Verification Model
3. Verification
    - **Model-checking**: checking properties which represent desired behaviors

Summary

1. Introduction
   - Hypermedia Document Design
   - Design Solutions
   - Main Challenges
2. The Proposed Design Method
   - Design Method
   - Transformation Step: from Designer Representation to Verification Representation
3. Formal Verification
   - Observers
   - Verification
   - Verification in Practice
4. Toolchain
   - Toolchain Associated to the Proposed Method
5. Conclusions
   - Performance Analyze
   - Conclusions

Main Challenges

1- Time Relationships Verification

Detection of undesirable behaviors originated from temporal relationships, and also remote control actions.

2- Spatial Relationships Verification

Guarantee of media display on the appropriate presentation region.

3- Live Editing

Verification on-the-fly with admissible response time.

4- Application Design Facilities

Friendly environment for designers without expertise on formal models.

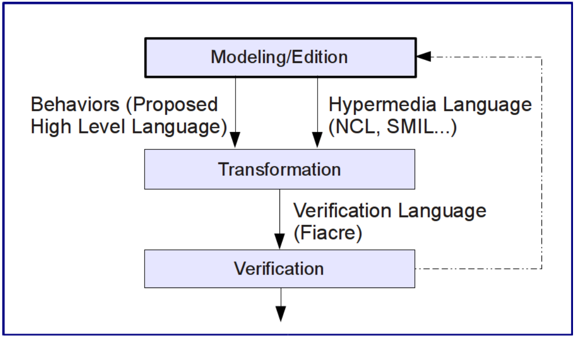
Summary

1. Introduction
   - Hypermedia Document Design
   - Design Solutions
   - Main Challenges
2. The Proposed Design Method
   - Design Method
   - Transformation Step: from Designer Representation to Verification Representation
3. Formal Verification
   - Observers
   - Verification
   - Verification in Practice
4. Toolchain
   - Toolchain Associated to the Proposed Method
5. Conclusions
   - Performance Analyze
   - Conclusions

Design Method

1- Modeling Step

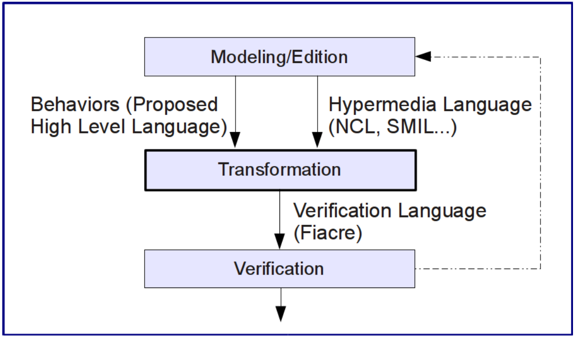
- Application written in hypermedia languages (NCL or SMIL).
- Desired Behaviors written in High Level Property Language.


Modeling/Edition
Behaviors (Proposed High Level Language)
Hypermedia Language (NCL, SMIL...)
Transformation
Verification Language (Fiacre)
Verification

Design Method

2- Automatic Transformation Step Using a MDE Approach

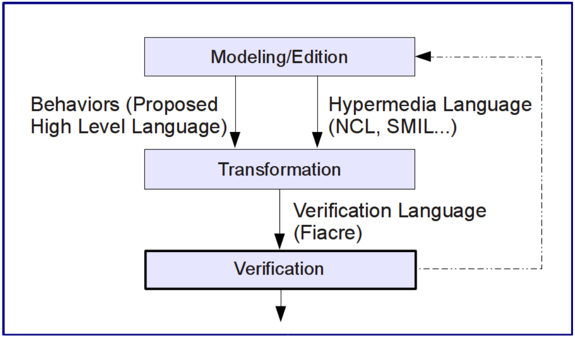
- **From** Hypermedia Application **to** Formal Verification Language (FIACRE).
- **From** High Level Property Language **to** LTL formula and FIACRE Observers.


Modeling/Edition
Behaviors (Proposed High Level Language)
Hypermedia Language (NCL, SMIL...)
Transformation
Verification Language (Fiacre)
Verification

Design Method

3- Verification Step (Model-checking Principles)

- Unsatisfied property → counterexample (sequence of actions corresponding to the non-satisfaction of the property).
- Counterexample helps the designer to fix the application errors.


Modeling/Edition
Behaviors (Proposed High Level Language)
Hypermedia Language (NCL, SMIL...)
Transformation
Verification Language (Fiacre)
Verification

Summary

1 Introduction
- Hypermedia Document Design
- Design Solutions
- Main Challenges

2 The Proposed Design Method
- Design Method
- Transformation Step: from Designer Representation to Verification Representation

3 Formal Verification
- Observers
- Verification
- Verification in Practice

4 Toolchain
- Toolchain Associated to the Proposed Method

5 Conclusions
- Performance Analyze
- Conclusions

Transformation Step

Transformation Rules: **From** Designer Representation **to** Verification Representation

From Hypermedia Language

1. Media (dynamics of the media)
2. Link between Medias
3. Possible User Interactions
4. High Level Property
5. High Level Property

To FIACRE Language

1. Fiacre Process
2. Fiacre Glue Process
3. Fiacre Remote Control Process
4. Fiacre Observer with time + LTL formula
5. LTL formula

Summary

1 Introduction
- Hypermedia Document Design
- Design Solutions
- Main Challenges

2 The Proposed Design Method
- Design Method
- Transformation Step: from Designer Representation to Verification Representation

3 Formal Verification
- Observers
- Verification
- Verification in Practice

4 Toolchain
- Toolchain Associated to the Proposed Method

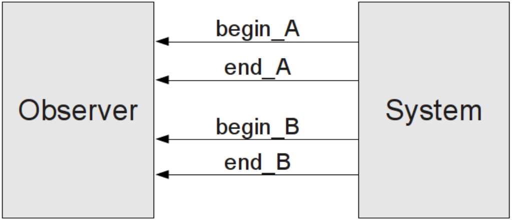
5 Conclusions
- Performance Analyze
- Conclusions

Verification

Behaviors can be checked in two ways:

- **LTL formulas**
  - when they consider only occurrence of events
- **Observers and LTL formulas**
  - when they measure the elapsed time between events
  - when they consider the cause of an event

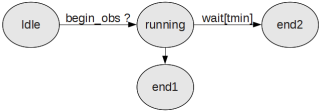
Observers capture events occurring in the Hypermedia System



Observers

Temporal Observer:

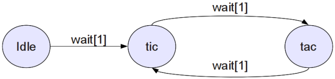
- The figure shows the basic observer which identifies the elapsed time between the arrival of the *begin_obs* and *end_obs* messages:
  1. end1: $elapsed\ time < t_{min}$
  2. end2: $t_{min} \leq elapsed\ time$



Observers

Global Time Observer- identifies a precise time when something occur.

- Aiming discretize the passage of time, the observer changes its state every second.
- Adopted in the analysis of counterexamples.



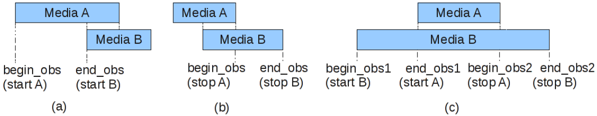
Summary

1. Introduction
   - Hypermedia Document Design
   - Design Solutions
   - Main Challenges
2. The Proposed Design Method
   - Design Method
   - Transformation Step: from Designer Representation to Verification Representation
3. Formal Verification
   - Observers
   - Verification
   - Verification in Practice
4. Toolchain
   - Toolchain Associated to the Proposed Method
5. Conclusions
   - Performance Analyze
   - Conclusions

Time Verification

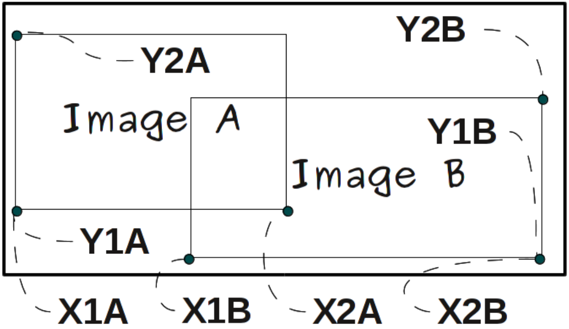
Only LTL formulas/Observers and LTL formulas

- **Intramedia relationships**- checks exhibition and time limits
- **Intermedia relationships**- checks all Allen's relationships, as:
    - (a) *B*-start **after** *A*-start
    - (b) *B*-stop **after** *A*-stop
    - (c) *A* **overlapping** *B*



Spatial Verification

- **Spatial** - checks full or partial spatial overlap of object or screen regions



Summary

1 Introduction
- Hypermedia Document Design
- Design Solutions
- Main Challenges

2 The Proposed Design Method
- Design Method
- Transformation Step: from Designer Representation to Verification Representation

3 Formal Verification
- Observers
- Verification
- Verification in Practice

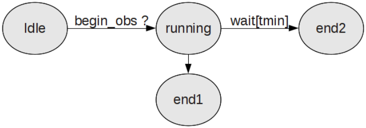
4 Toolchain
- Toolchain Associated to the Proposed Method

5 Conclusions
- Performance Analyze
- Conclusions

Verification in Practice

Application "Live Longer"

- **Erroneous behavior**- menu displayed out of human visual perception
- **Intramedia Property**- when presented, media $menu\_Dish1$ always remains visible for a minimum time observable by perceptible human vision
  - The property to check the vision time, is represented by ob_menu_Dish1 observer

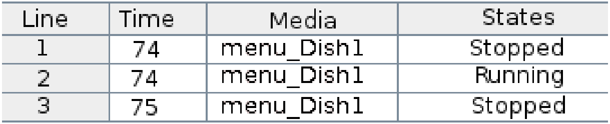


- the observer's behavior is verified by LTL formula:
- $\Box(ob\_menu\_dish1\_running \implies (\neg(\Diamond(ob\_menu\_dish1\_end1))))$
- **The result is "False"**

Counterexample

Counterexample media *menu_dish*1

- The time between **running** and **stopped** states is less than the minimum required.
- **Erroneous behavior**- *menu_dish*1 displayed out of human visual sense



Line	Time	Media	States
1	74	menu_Dish1	Stopped
2	74	menu_Dish1	Running
3	75	menu_Dish1	Stopped

- This table is part of a graphical interface, generated after the verification process.
- The Time column is generated from the Global Time Observer.

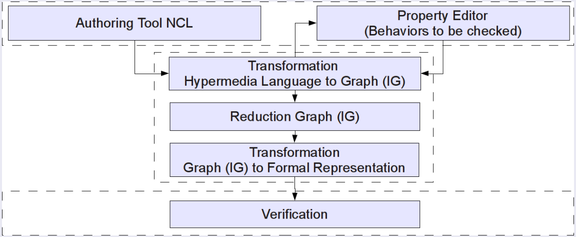
Summary

1. Introduction
   - Hypermedia Document Design
   - Design Solutions
   - Main Challenges
2. The Proposed Design Method
   - Design Method
   - Transformation Step: from Designer Representation to Verification Representation
3. Formal Verification
   - Observers
   - Verification
   - Verification in Practice
4. Toolchain
   - Toolchain Associated to the Proposed Method
5. Conclusions
   - Performance Analyze
   - Conclusions

The Proposed Toolchain

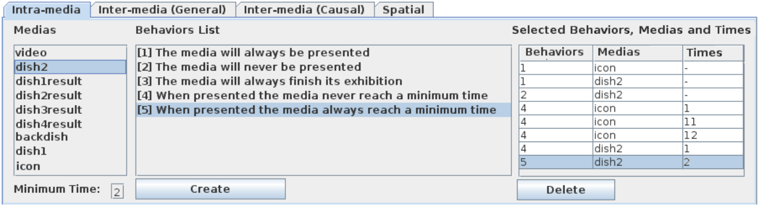
Toolchain

1. Modeling/Edition
   - **Authoring Tool** and **Property Editor**
2. Transformation
   - **From Hypermedia Language to Intermediary Graph (IG)**
   - **Reduction IG Graph**
   - **From IG Graph to Formal Representations**
3. Verification


Authoring Tool NCL
Property Editor
(Behaviors to be checked)
Transformation
Hypermedia Language to Graph (IG)
Reduction Graph (IG)
Transformation
Graph (IG) to Formal Representation
Verification

1- Property Editor

- *Graphical User Interface* (GUI) assist the designer specification of types of behaviors:
  - Intra-media
  - Inter-media
  - Causal
  - Spatial



Medias

- video
- dish2
- dish1result
- dish2result
- dish3result
- dish4result
- backdish
- dish1
- icon

Behaviors List

[1] The media will always be presented
[2] The media will never be presented
[3] The media will always finish its exhibition
[4] When presented the media never reach a minimum time
[5] When presented the media always reach a minimum time

Selected Behaviors, Medias and Times

Behaviors	Medias	Times
1	icon	-
1	dish2	-
2	dish2	-
4	icon	1
4	icon	11
4	icon	12
4	dish2	1
5	dish2	2

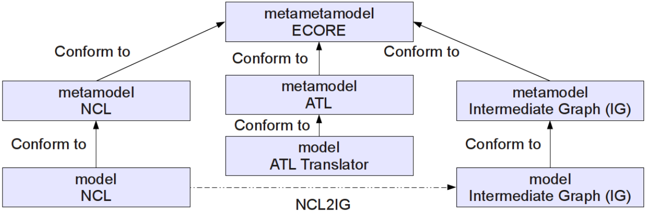
Minimum Time: 2

2- Transformation Hypermedia to IG Graph

IG Graph:

- Allows use of graph theory in the reduction process
- Add new Hypermedia language to the verification chain

- MDE Transformation model-to-model (M2M): transforms from NCL application to Intermediary Graph (IG)
- Transformation rules coded in ATL language

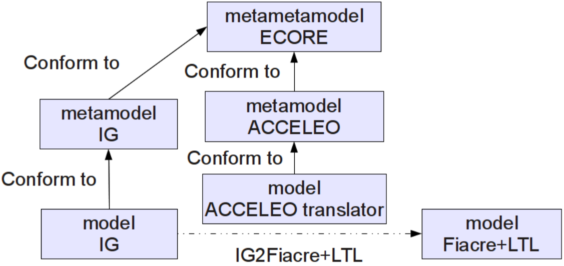


3- Reducing IG Graph

- Goal: reduce the computational cost of the verification process during the **live design**.
- Receives as inputs IG and a set of properties
  - performs reduction for each media and property associated
  - preserves the relevant parts of this graph for checking the desired properties
- Developed in Java

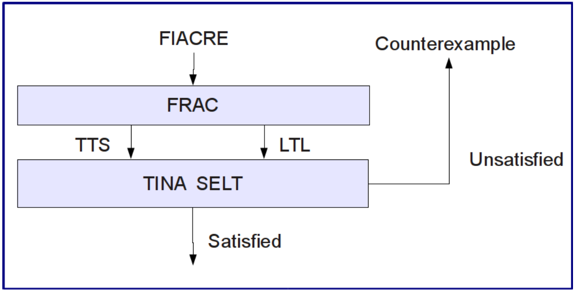
4- Transformation IG Graph to Formal Representation

- MDE Transformation model-to-text (M2T): transforms from IG Graph to FIACRE Model
- MDE Transformation model-to-text (M2T): transforms from High Level Properties to FIACRE Properties (LTL)
- Transformation rules coded in ACCELEO language



5- Verification

- The code in Fiacre is compiled by the FRAC tool generating an equivalent code in TTS and LTL
- SELT, the model checker tool of the TINA toolbox
- SELT allows to verify formulas written in LTL
    - When the formula is unsatisfied, a counterexample is generated to help the designer



Summary

1. Introduction
    - Hypermedia Document Design
    - Design Solutions
    - Main Challenges
2. The Proposed Design Method
    - Design Method
    - Transformation Step: from Designer Representation to Verification Representation
3. Formal Verification
    - Observers
    - Verification
    - Verification in Practice
4. Toolchain
    - Toolchain Associated to the Proposed Method
5. Conclusions
    - Performance Analyze
    - Conclusions

Performance Analyze

Verification of previous application "Live Longer". In all cases, the reduction resulted in a decrease in the size of the model, as well as lower response time

Description	States	Transitions	Time
Complete Model	26448	94454	18 seconds
Reduced Model	18576	66198	3 seconds
Complete Model (1 Observer)	33678	120688	29 seconds
Reduced Model (1 Observer)	22830	79928	4 seconds
Complete Model (2 Observers)	44105	161067	37 seconds
Reduced Model (2 Observers)	29017	101235	5 seconds

Summary

1. Introduction
    - Hypermedia Document Design
    - Design Solutions
    - Main Challenges
2. The Proposed Design Method
    - Design Method
    - Transformation Step: from Designer Representation to Verification Representation
3. Formal Verification
    - Observers
    - Verification
    - Verification in Practice
4. Toolchain
    - Toolchain Associated to the Proposed Method
5. Conclusions
    - Performance Analyze
    - Conclusions

Conclusions

- Proposal and validation of a Design Method
- Development and test of toolchain supporting this Design Method based on MDE
  - Guarantee of coherence between hypermedia model and formal model - **MDE**
  - Issues for Hypermedia Document
    - requirement to verify (temporal, causal and spatial)
    - live design (by reduction, decreasing the computational cost)
    - facilities for designer without expertise in verification

Questions and Contacts

- Cristian Koliver
  - ckoliver@gmail.com
- Delcino Picinin Júnior
  - picinin@gmail.com
- Jean-Marie Farines
  - j.m.farines@ufsc.br
- Celso Alberto Saibel Santos
  - celsoalbertosaibelsantos@gmail.com

Verifying Hypermedia Applications by Using a MDE Approach

Cristian Koliver - Federal University of Santa Catarina,
Delcino Picinin Júnior - Federal Institute of Santa Catarina,
Celso A. S. Santos - Federal University of Espirito Santo and
Jean-Marie Farines - Federal University of Santa Catarina

BRASIL

Valencia, 29 september 2014